

TUGAS AKHIR - KI141502

PERANCANGAN DAN IMPLEMENTASI GAME TOWER DEFENSE: THE LEGEND GUARDIANS BERBASIS ANDROID

MUHAMMAD ABDURROHMAN
NRP 05111240007004

Dosen Pembimbing
Imam Kuswardayan, S.Kom., M.T.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



TUGAS AKHIR - KI141502

PERANCANGAN DAN IMPLEMENTASI GAME TOWER DEFENSE: THE LEGEND GUARDIANS BERBASIS ANDROID

**MUHAMMAD ABDURROHMAN
NRP 05111240007004**

**Dosen Pembimbing
Imam Kuswardayan, S.Kom., M.T.**

**DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

(Halaman Ini Sengaja dikosongkan)



FINAL PROJECT - KI141502

DESIGN AND IMPLEMENTATION TOWER DEFENSE GAME: THE LEGEND GUARDIANS BASED ON ANDROID

**MUHAMMAD ABDURROHMAN
NRP 05111240007004**

**Dosen Pembimbing
Imam Kuswardayan, S.Kom., M.T.**

**DEPARTMENT OF INFORMATICS
Faculty of Information Technology and Communication
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

(Halaman Ini Sengaja dikosongkan)

LEMBAR PENGESAHAN

PERANCANGAN DAN IMPLEMENTASI GAME TOWER DEFENSE: THE LEGEND GUARDIANS BERBASIS ANDROID

Tugas Akhir

Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Interaksi, Grafika, dan Seni
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

Muhammad Abdurrohman

NRP. 051112406010064

Disetujui oleh Dosen Pembimbing Tugas Akhir

Imam Kuswardayan, S.Kom., M.

NIP: 19761215 200312 1 001



(pembimbing)

**SURABAYA
JULI, 2018**

(Halaman Ini Sengaja dikosongkan)

PERANCANGAN DAN IMPLEMENTASI GAME TOWER DEFENSE: THE LEGEND GUARDIANS BERBASIS ANDROID

Nama Mahasiswa : Muhammad Abdurrohman
NRP : 0511240007004
Departemen : Informatika - Fakultas Teknologi
Informasi dan Komunikasi - ITS
Dosen Pembimbing : Imam Kuswardayan, S.Kom., M.T.

ABSTRAK

Kemajuan teknologi informasi menjadikan dunia gaming berkembang secara pesat. Berbagai macam genre, cara bermain, dan fitur-fitur lainnya didalam game terus berkembang dan menjadi sangat beragam. Salah satu genre game yang cukup banyak diminati adalah genre tower defense. Tower Defense (TD) adalah sebuah subgenre dari game strategi yang bertujuan untuk mempertahankan wilayah kekuasaan atau harta pemain dari musuh. Terdapat beberapa macam cara bermain game dengan genre tower defense ini, diantaranya adalah dengan cara memilih pasukan dan meletakkannya di tempat yang tepat agar wilayah pemain tidak dihancurkan oleh musuh. Terdapat juga cara bermain dengan memilih pasukan untuk bertahan dari serangan musuh sekaligus menghancurkan wilayah musuh.

Pada usulan tugas akhir ini, game yang akan dibuat adalah game dengan genre tower defense yang akan dimainkan melawan bot/komputer. Bot akan dirancang sedemikian rupa agar memiliki tingkat kesulitan yang meningkat seiring bertambahnya level. Cara bermain yang digunakan adalah dengan memilih pasukan untuk bertahan dari serangan musuh sekaligus menghancurkan wilayah musuh untuk memenangkan permainan. Dengan sistem seperti ini, akan membuat permainan menjadi lebih strategis dan menarik.

Dari perancangan dan implementasi game tower defense: The Legend Guardians berbasis android ini, diharapkan banyak orang mengerti bagaimana aturan main, skenario, dan tingkat kesulitannya serta dapat meningkatkan pola pikir strategis bagi para pemain sekaligus memberikan hiburan.

Kata kunci: Android, Bot, Strategi, Tower Defense

DESIGN AND IMPLEMENTATION TOWER DEFENSE GAME: THE LEGEND GUARDIANS BASED ON ANDROID

Student Name : Muhammad Abdurrohman
NRP : 0511240007004
Department : Informatics – Faculty of Information
Technology and Communication - ITS
Advisor : Imam Kuswardayan, S.Kom., M.T.

ABSTRACT

The advances in information technology makes the world of gaming is growing rapidly. Various genres, ways of playing, and other features in the game continue to grow and become very diverse. One of the genre of games that are quite popular is the tower defense genre. Tower Defense (TD) is a subgenre of strategy games aimed at defending the territory or treasure of players from the enemy. There are several ways to play the game with this tower defense genre, such as by choosing troops and putting them in the right place so that the player's territory is not destroyed by the enemy. There is also a way of playing by choosing troops to defend against enemy attacks while destroying enemy territory.

In this final project proposal, the game to be created is a game with the tower defense genre that will be played against the bot / computer. Bot will be designed in such a way as to have a level of difficulty that increases with increasing levels. How to play that used in this game is to choose troops to survive from enemy attacks and destroy enemy territory to win the game. With a system like this, it will make the game more strategic and interesting. Players can play and improve as many strategy as they can.

From the design and implementation of this tower defense game: The Legend Guardians based on android, it is expected that many people understand how the rules of the game, scenario, and

level of difficulty and can improve the strategic mindset for the players while providing entertainment.

Keywords: Android, Bot, Strategy, Tower Defense

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillah Rabbil ‘Alamin, segala puji bagi Allah SWT yang selalu mencurahkan limpahkan karunia dan rahmat-Nya kepada penulis, sehingga tugas akhir dengan judul “Perancangan dan Implementasi *Game Tower Defense: The Legend Guardians* Berbasis Android” ini dapat selesai walau tidak sesuai dengan waktu yang telah ditentukan. Pengerjaan tugas akhir ini menjadi sebuah sarana untuk penulis mematangkan ilmu yang telah didapatkan selama menempuh pendidikan di kampus perjuangan Institut Teknologi Sepuluh Nopember Surabaya, khususnya dalam disiplin ilmu Informatika. Penyelesaian buku tugas akhir ini tidak terlepas dari bantuan dan dukungan berbagai pihak, oleh karena itu penulis ingin mengucapkan terima kasih kepada:

1. Allah SWT. dan junjungan Nabi Muhammad SAW.
2. Kedua orang tua dan keluarga yang telah memberikan dukungan moral dan material serta doa yang terus menerus demi kesuksesan penulis.
3. Bapak Imam Kuswardayan selaku dosen pembimbing yang telah membantu dan membimbing penulis dalam menyelesaikan tugas akhir ini.
4. Ibu Dr. Chastine Fatichah, M.Kom. selaku dosen wali penulis, yang telah membantu di setiap awal semester dan selalu memudahkan perkuliahan penulis selama di kampus Informatika ITS.
5. Bapak Darlis Herumurti, S.Kom, M.Kom. selaku Kepala Departemen Informatika ITS, Bapak Radityo Anggoro, S.Kom, M.Sc. selaku koordinator Tugas Akhir, dan segenap dosen Informatika yang telah memberikan ilmunya serta staf karyawan Departemen Informatika ITS yang telah memberikan bantuan demi kelancaran admisnistrasi penulis selama kuliah.

6. Teman-teman CSSMoRA ITS angkatan 2012 yang telah berjuang bersama-sama sejak matrikulasi hingga saat ini di kampus perjuangan ITS.
7. Adik-adik CSSMoRA ITS angkatan 2014 kebawah yang telah memberikan semangat penulis dalam menyelesaikan tugas akhir.
8. Teman-teman Departemen Informatika angkatan 2012 yang telah mewarnai masa perkuliahan penulis.
9. Semua pihak yang sudah membantu dan tidak bisa penulis tuliskan satu persatu, maaf dan terimakasih.

Penulis menyadari bahwa tugas akhir ini masih memiliki banyak kekurangan. Dengan demikian, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan kedepannya.

Surabaya, Juli 2018

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
DAFTAR KODE SUMBER	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	2
1.6 Metodologi	3
1.7 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA.....	7
2.1 <i>Game Tower Defense</i>	7
2.2 <i>Game Tower Defense</i> Serupa	9
2.2.1 <i>Line Rangers</i>	10
2.2.2 <i>Age of Empires</i>	12
2.2.3 <i>Chomical</i>	13
2.2.4 <i>Vector Tower Defense 2</i>	14
2.2.5 <i>Emotion</i>	16
2.3 Android.....	17
2.4 <i>Game Engine Unity</i>	17
BAB III ANALISIS DAN PERANCANGAN.....	19
3.1 Analisis Sistem	19
3.2 Perancangan <i>Game</i>	19
3.2.1 Deskripsi Umum <i>Game</i>	20
3.2.2 Aturan Permainan	21
3.2.3 Skenario	22
3.2.4 Tingkat Kesulitan.....	23

3.3	Perancangan Antarmuka	24
3.3.1	Rancangan Antarmuka Awal	24
3.3.2	Rancangan Antarmuka Menu Utama	25
3.3.3	Rancangan Antarmuka <i>Tutorial</i>	26
3.3.4	Rancangan Antarmuka <i>Select Level</i>	26
3.3.5	Rancangan Antarmuka <i>Game</i>	27
BAB IV	IMPLEMENTASI	29
4.1	Lingkungan Implementasi	29
4.2	Implementasi Permainan	29
4.2.1	Implementasi Antarmuka Awal	29
4.2.2	Implementasi Antarmuka Menu Utama	31
4.2.3	Implementasi Antarmuka <i>Tutorial</i>	33
4.2.4	Implementasi Antarmuka <i>Select Level</i>	34
4.2.5	Implementasi Antarmuka <i>Game</i>	35
BAB V	PENGUJIAN DAN EVALUASI	43
5.1	Lingkungan Uji Coba	43
5.2	Pengujian Fungsionalitas	43
5.2.1	Uji Coba Pada Antarmuka Awal	44
5.2.2	Uji Coba Pada Antarmuka Menu Utama	45
5.2.3	Uji Coba Pada Antarmuka <i>Tutorial</i>	46
5.2.4	Uji Coba Pada Antarmuka <i>Select Level</i>	47
5.2.5	Uji Coba Pada Antarmuka <i>Game</i>	50
5.3	Pengujian Alur Permainan	53
5.4	Pengujian Tingkat Kesulitan	56
5.5	Pengujian Usability	57
5.6	Evaluasi	57
BAB VI	KESIMPULAN DAN SARAN	61
6.1.	Kesimpulan	61
6.2.	Saran	61
DAFTAR	PUSTAKA	63
LAMPIRAN	65
BIODATA	PENULIS	71

DAFTAR GAMBAR

Gambar 2.1 Antarmuka awal <i>game Line Rangers</i>	10
Gambar 2.2 Antarmuka menu <i>game Line Rangers</i>	11
Gambar 2.3 Antarmuka <i>gameplay game Line Rangers</i>	11
Gambar 2.4 Antarmuka awal <i>game Age of Empires</i>	12
Gambar 2.5 Antarmuka <i>gameplay game Age of Empires</i>	13
Gambar 2.6 Antarmuka awal <i>game Chomical</i>	14
Gambar 2.7 Antarmuka <i>game Vector Tower Defense 2</i>	15
Gambar 2.8 Antarmuka <i>game Emotion</i>	16
Gambar 3.1 Rancangan antarmuka awal	24
Gambar 3.2 Rancangan antarmuka menu utama	25
Gambar 3.3 Rancangan antarmuka <i>tutorial</i>	26
Gambar 3.4 Rancangan antarmuka <i>select level</i>	27
Gambar 3.5 Rancangan antarmuka <i>game</i>	28
Gambar 4.1 Implementasi antarmuka awal	30
Gambar 4.2 Implementasi antarmuka menu utama	31
Gambar 4.3 Implementasi antarmuka <i>tutorial</i>	33
Gambar 4.4 Implementasi antarmuka <i>select level</i>	34
Gambar 4.5 Implementasi antarmuka <i>game</i>	36
Gambar 5.1 Uji coba antarmuka <i>tutorial</i>	46
Gambar 5.2 Skenario uji coba antarmuka <i>select level</i>	48
Gambar 5.3 Skenario uji coba antarmuka <i>game</i>	50
Gambar 5.4 Skenario <i>game paused</i>	53
Gambar 5.5 Kondisi menang <i>game</i>	55
Gambar 5.6 Kondisi kalah <i>game</i>	55
Gambar 8.1 Kuesioner <i>responden 1</i>	65
Gambar 8.2 Kuesioner <i>responden 2</i>	66
Gambar 8.3 Kuesioner <i>responden 3</i>	67
Gambar 8.4 Kuesioner <i>responden 4</i>	68
Gambar 8.5 Kuesioner <i>responden 5</i>	69

(Halaman ini Sengaja dikosongkan)

DAFTAR TABEL

Tabel 3.1 Skenario <i>Hero</i>	22
Tabel 3.2 Skenario <i>Monster</i>	22
Tabel 3.3 Tingkat Kesulitan	23
Tabel 5.1 Hasil uji coba antarmuka awal	44
Tabel 5.2 Hasil uji coba antarmuka menu utama	45
Tabel 5.3 Hasil uji coba antarmuka <i>tutorial</i>	47
Tabel 5.4 Hasil uji coba antarmuka <i>select level</i>	48
Tabel 5.5 Hasil uji coba antarmuka <i>game</i>	51
Tabel 5.6 Hasil pengujian alur permainan	53
Tabel 5.7 Hasil pengujian tingkat kesulitan	56
Tabel 5.8 Rekapitulasi hasil kuesioner	58
Tabel 5.9 Evaluasi hasil uji coba	59
Tabel 5.10 Rekapitulasi evaluasi	60

(Halaman ini Sengaja dikosongkan)

DAFTAR KODE SUMBER

Kode Sumber 4.1 Fungsi antarmuka awal.....	30
Kode Sumber 4.2 Fungsi antarmuka menu utama.....	33
Kode Sumber 4.3 Fungsi antarmuka <i>select level</i>	35
Kode Sumber 4.4 Fungsi antarmuka <i>game</i>	39
Kode Sumber 4.5 Fungsi pada <i>hero</i>	42

(Halaman Ini Sengaja dikosongkan)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi informasi menjadikan dunia *gaming* berkembang secara pesat. Berbagai macam *genre*, cara bermain, dan fitur-fitur lainnya didalam *game* terus berkembang dan menjadi sangat beragam. Salah satu *genre game* yang cukup banyak diminati adalah *genre tower defense*. *Tower defense* adalah sebuah *subgenre* dari *game* strategi yang bertujuan untuk mempertahankan wilayah kekuasaan atau harta pemain dari musuh. Terdapat beberapa macam cara bermain *game* dengan *genre tower defense* ini, diantaranya adalah dengan cara memilih pasukan dan meletakkannya di tempat yang tepat agar wilayah pemain tidak dihancurkan oleh musuh. Terdapat juga cara bermain dengan memilih pasukan untuk bertahan dari serangan musuh sekaligus menghancurkan wilayah musuh.

Pada usulan tugas akhir ini, *game* yang akan dibuat adalah *game* dengan *genre tower defense* yang akan dimainkan melawan *bot/komputer*. *Bot* akan dirancang sedemikian rupa agar memiliki tingkat kesulitan yang meningkat seiring bertambahnya *level*. Cara bermain yang digunakan adalah dengan memilih pasukan untuk bertahan dari serangan musuh sekaligus menghancurkan wilayah musuh untuk memenangkan permainan. Dengan sistem seperti ini, akan membuat permainan menjadi lebih strategis dan menarik.

Dari perancangan dan implementasi *game tower defense: The Legend Guardians* berbasis Android ini, diharapkan banyak orang mengerti bagaimana aturan main, skenario, dan tingkat kesulitannya serta dapat meningkatkan pola pikir strategis bagi para pemain sekaligus memberikan hiburan.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut :

1. Bagaimana merancang dan mengimplementasikan aturan main, skenario, dan tingkat kesulitan pada *game The Legend Guardians* berbasis Android?
2. Bagaimana merancang dan mengimplementasikan *game The Legend Guardians* menggunakan *game engine* Unity?

1.3 Batasan Masalah

Batasan masalah dari pembuatan tugas akhir ini terdapat pada *game engine* yang digunakan untuk merancang dan mengimplementasikan *game*. *Game engine* yang akan digunakan adalah Unity versi 2017.2.0f3.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah membuat *game tower defense: The Legend Guardians* berbasis Android.

1.5 Manfaat

Manfaat dari hasil pembuatan tugas akhir ini adalah sebagai berikut :

1. Mengetahui rancangan dan implementasi aturan main, skenario, dan tingkat kesulitan pada *game The Legend Guardians* berbasis Android.
2. Mengetahui rancangan dan implementasi *game The Legend Guardians* menggunakan *game engine* Unity.
3. Meningkatkan pola pikir yang strategis bagi para pemain.
4. Memberikan hiburan bagi para pemain.

1.6 Metodologi

1. Penyusunan proposal tugas akhir

Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan ini terdiri dari hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat dari hasil pembuatan tugas akhir. Selain itu, dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Subbab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula subbab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

2. Studi literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan *game* ini, yaitu mengenai *game* pada *platform* Android, *game engine* Unity, dan bahasa pemrograman C#.

3. Analisis dan desain perangkat lunak

Fitur yang terdapat pada *game* ini adalah sebagai berikut :

1. Jumlah pemain dua (pemain dan *bot*).
2. Grafik 2 dimensi.
3. Jenis *game* adalah *tower defense*.

4. Implementasi perangkat lunak

Game yang akan dibuat adalah *game tower defense* yang dapat berjalan di perangkat Android. Dalam membuat *game* ini, penulis menggunakan *game engine* Unity versi 2017.2.0f3 dengan bahasa pemrograman C#. Untuk antarmuka *game* didesain menggunakan aplikasi Corel Draw X7, Adobe Illustrator CS6, dan Adobe Photoshop CS6.

5. Pengujian dan evaluasi

Terdapat beberapa macam pengujian yang akan dilakukan pada game yang akan dibuat ini. Pengujian-pengujian tersebut akan dijelaskan sebagai berikut :

1. Pengujian *blackbox*

Pengujian *blackbox* adalah pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, penguji dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. Pengujian ini dilakukan untuk menguji apakah proses kinerja *game* ini sudah sesuai dengan kebutuhan pemain atau tidak.

2. Pengujian alur permainan

Pengujian alur permainan adalah pengujian yang dilakukan untuk mengetahui apakah setiap *level* pada permainan memungkinkan untuk diselesaikan. Pada pengujian ini terdapat dua skenario, yaitu skenario menang dan skenario kalah.

3. Pengujian tingkat kesulitan

Pengujian tingkat kesulitan adalah pengujian yang dilakukan untuk mengetahui apakah tingkat kesulitan pada setiap *level* semakin meningkat mulai dari *level* rendah hingga *level* tinggi. Akan terdapat beberapa parameter yang menjadi tolok ukur dari tingkat kesulitan setiap *level*.

4. Pengujian *usability*

Pengujian *usability* dilakukan dengan cara melakukan survei ke pemain di sekitar lingkungan Informatika ITS. Survei dilakukan untuk mengukur nilai dalam hal antarmuka, performa, dan hiburan/*fun* dari *game* yang dibuat.

Setelah berbagai macam pengujian yang dilakukan, maka akan dilakukan evaluasi terhadap macam-macam pengujian

tersebut. Evaluasi akan menilai hasil dari pengujian-pengujian yang dilakukan kemudian mengkategorikannya kepada berhasil atau tidak dengan membuat rekapitulasinya.

6. Penyusunan buku tugas akhir

Pada tahapan ini dilakukan penyusunan buku yang menjelaskan dasar teori dan metode yang digunakan dalam pembuatan tugas akhir ini serta memuat dokumentasi mengenai perancangan dan hasil dari implementasi *game* yang telah dibuat.

1.7 Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk menjelaskan proses-proses dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir ini terdiri atas beberapa bagian seperti berikut ini:

Bab I Pendahuluan

Bab ini berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu, rumusan permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Tinjauan Pustaka

Bab ini berisi penjelasan secara detil mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan tugas akhir ini.

Bab III Analisis dan Perancangan

Bab ini membahas tahap analisis permasalahan dan perancangan dari *game* yang akan dibangun. Analisis permasalahan membahas permasalahan-permasalahan yang diangkat dalam pengerjaan tugas akhir.

Perancangan yang dibuat adalah perancangan aturan main, skenario, dan tingkat kesulitan *game* yang dibuat.

Bab IV Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Pada bab ini akan dijelaskan secara detil dari proses implementasi tersebut.

Bab V Pengujian Dan Evaluasi

Bab ini membahas pengujian dengan beberapa metode, yaitu pengujian alur permainan, pengujian tingkat kesulitan, pengujian *blackbox* yang berfokus pada spesifikasi fungsional dari *game*, dan pengujian *usability* yang dilakukan dengan cara melakukan survei ke para pemain.

Bab VI Kesimpulan Dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan *game* kedepannya.

BAB II

TINJAUAN PUSTAKA

Bab ini berisi penjelasan teori-teori yang berkaitan dengan metode yang diajukan pada pengimplementasian *game*. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap sistem yang dibuat dan berguna sebagai penunjang dalam pengembangan *game*.

2.1 *Game Tower Defense*

Tower Defense (TD) adalah sebuah *subgenre* dari *game* strategi di mana tujuannya adalah untuk mempertahankan wilayah kekuasaan atau harta pemain dari musuh. Pada umumnya, *game* ini dimainkan dengan cara menempatkan struktur *defensif/tower* di sepanjang jalur serangan musuh. *Tower* ini dibangun dengan berbagai fungsi yang secara otomatis memblokir, menghambat serangan, dan menghancurkan musuh. Pemilihan dan penempatan posisi *tower* yang tepat adalah strategi terpenting untuk *genre game* ini.

Sejarah perkembangan *genre game tower defense* bermula dari zaman keemasan *genre game arcade* pada tahun 1980-an. *Game Space Invaders* adalah sebuah *game* yang rilis pada tahun 1978. Pada *game* ini, terdapat objek-objek yang digunakan untuk mempertahankan wilayah pemain untuk melawan *wave/gelombang* serangan musuh yang masuk ke wilayah pemain tersebut. Pada *game* ini, dilengkapi dengan fitur *shield/perisai* yang secara strategis dapat digunakan untuk menghalangi serangan musuh kepada pemain dan membantu pemain mempertahankan wilayah pemain meskipun fitur ini pada awalnya tidak spesifik dibuat untuk mempertahankan wilayah pemain tersebut.

Pada tahun 1980 sebuah *game* bernama *Missile Command* dirilis. *Game* ini memperkenalkan sebuah elemen strategi. Didalam *game*, pemain dapat menghalangi *missile* yang masuk ke wilayah pemain dan juga terdapat beberapa jalur serangan di setiap *wave* serangan musuh. *Game* ini juga adalah *game* yang pertama menggunakan perangkat penunjuk, *trackball*, yang digunakan pemain

dalam memainkan *game* tersebut. Fitur ini adalah fitur yang nantinya akan berkembang menjadi *mouse* pada komputer. Selain itu, pada *game* ini juga satu-satunya target penyerangan adalah *base* musuh, bukanlah karakter musuh tertentu. Untuk alasan ini, beberapa kalangan menganggapnya sebagai *game* sejati pertama didalam *genre tower defense* ini.

Pada perkembangannya, muncul juga *game* yang bernama *Defender* yang rilis pada tahun 1981 dan *Choplifter* yang rilis pada tahun 1982. Kedua *game* ini dibuat dengan memperbaiki kekurangan-kekurangan yang ada pada *game Missile Command*. Bahkan pada kedua *game* ini, tujuan utama dari permainan diubah untuk mempertahankan *non-player items*. Mempertahakan *non-player items* ini adalah kunci dari permainan kedua *game* ini.

Setelah berbagai macam *game* ini bermunculan, pada tahun 1990 dirilis sebuah *game* yang bernama *Rampart*. *Game* ini lah yang mengenalkan fitur untuk pemain dapat memasang sebuah struktur *defensif* yang secara otomatis menyerang musuh-musuh yang datang. *Game* ini dilengkapi juga fase-fase dalam pembangunan struktur *defensif* tersebut, dalam mempertahankannya, dan dalam melakukan *repair* terhadapnya. Hal ini adalah elemen-elemen utama yang nantinya akan digunakan dan dikembangkan dalam *genre game* ini. *Game* ini juga adalah *game* pertama yang memiliki fitur *multiplayer* pada *genre game* ini.

Pada tahun 2007 dan 2008, *genre game tower defense* ini menjadi fenomenal dikarenakan popularitas dari *genre game tower defense* dan *game RTS (Real Time Strategy)* yang semakin meningkat secara drastis. Dalam perkembangannya, banyak dari *game* dengan *genre* ini yang dilengkapi dengan *touchscreen interface* dan dimainkan sebagai *game mobile* seperti *Bloons TD 4* yang rilis pada tahun 2009 dan *Plants vs. Zombies* yang juga rilis pada tahun 2009. Kedua *game* tersebut juga sudah menjadi lebih kompleks karena memiliki fitur *turn-base gameplay*, *mini games*, dan berbagai macam fitur lainnya.

Dari berbagai jenis *game tower defense* yang telah berkembang, pada intinya semua *game* tersebut memiliki *basic gameplay* yang sama, yaitu :

1. Terdapat wilayah atau yang biasa disebut dengan *base* yang harus dipertahankan oleh pemain,
2. *Base* harus dapat bertahan dari beberapa *waves* serangan musuh, dan
3. Penempatan elemen-elemen *tower* di sepanjang jalur penyerangan musuh.

Hal-hal yang membedakan berbagai jenis dalam *genre game tower defense* tersebut adalah kemampuan pemain untuk dapat menempatkan atau membangun penghalang secara strategis di jalur penyerangan musuh. Terdapat juga *tower* yang dibangun untuk menghancurkan *base* musuh, fitur untuk *repair tower*, *upgrade tower*, dan berbagai macam kemampuan musuh yang beragam [1].

2.2 Game Tower Defense Serupa

Dalam pembuatan *game tower defense: The Legend Guardians* ini, terdapat 2 *games* yang menjadi inspirasi yaitu *Line Rangers* dan *Age of Empires*. Walaupun memiliki cara bermain yang berbeda, kedua *game* ini memiliki unsur *tower defense* dalam permainan. Cara bermain dengan memilih pasukan yang akan dikeluarkan sesuai *energy* yang sudah terkumpul ini menyerupai *game Line Rangers*. Sedangkan *game Age of Empires* memiliki berbagai macam strategi yang dapat dilakukan. Hal ini dapat diterapkan pada *game* yang akan dibuat. Selain itu, terdapat juga 3 *games* lainnya yang menjadi referensi tambahan agar *game* yang dibuat menjadi semakin menarik dan bervariasi. 3 *games* tersebut adalah *Chomical*, *Vector Tower Defense 2*, dan *Emotion Game*. Seperti pada 2 *games* yang menjadi inspirasi dalam pembuatan *game* ini, ketiga *games* ini pun memiliki unsur *tower defense* dalam permainan. Selain itu, ketiga *games* ini memiliki keunikan-keunikan tersendiri yang dipadukan dengan unsur *tower defense* sehingga menjadi *games* yang menarik dan dapat dijadikan referensi dalam pembuatan *game* ini.

2.2.1 *Line Rangers*

Line Rangers adalah sebuah *game* yang diluncurkan oleh Line Corporation pada tahun 2014. *Game* ini dapat dimainkan di perangkat *mobile*. Permainan yang sederhana namun adiktif ini memiliki karakter utama dari *icon-icon* aplikasi Line. Seiring perkembangannya, *game* ini menjadi semakin diminati karena



Gambar 2.1 Antarmuka awal *game Line Rangers* [2]

pertambahan berbagai macam karakter yang unik, berbagai *level* baru dengan tingkat kesulitan yang terus bertambah, dan *event-event* yang menarik. Antarmuka awal *game Line Rangers* seperti pada Gambar 2.1.

Pemain harus mengatur formasi pasukan sebaik mungkin sebelum memulai permainan. Terdapat berbagai macam pasukan yang memiliki peran yang berbeda-beda, diantaranya ada yang berperan sebagai *tanker*, *fighter*, *DPS*, dan sebagainya. Komposisi pasukan yang baik akan memudahkan pemain mengalahkan pemain yang lain dan memenangkan *level-level* yang ada. Antarmuka menu *game Line Rangers* seperti pada Gambar 2.2.

Game ini dimainkan di peta 2 dimensi horizontal dimana pemain dan musuh memiliki basis pertahanan masing-masing di kedua ujung arena. Seiring waktu, pemain menghasilkan mata uang

yang akan digunakan untuk membeli pasukan dan dikirim ke medan perang. Setiap pasukan memiliki kekuatan dan harga yang berbeda.

Setiap pasukan yang dikeluarkan akan berjalan menuju basis pertahanan musuh. Jika mereka menghadapi musuh yang menghalangi jalan mereka, maka mereka akan saling bertempur. Tujuan permainan ini adalah untuk menghancurkan pasukan musuh dan basis pertahanan musuh. Antarmuka *gameplay game Line Rangers* seperti pada Gambar 2.3 [2].



Gambar 2.2 Antarmuka menu *game Line Rangers* [2]

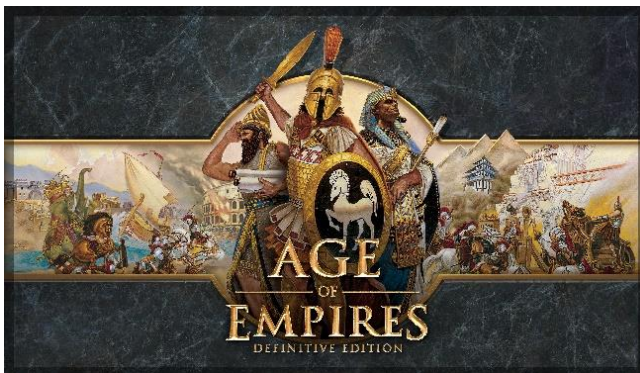


Gambar 2.3 Antarmuka *gameplay game Line Rangers* [2]

2.2.2 *Age of Empires*

Age of Empires adalah sebuah *game* dengan *genre real-time strategy* yang dikembangkan oleh Ensemble Studios dan diluncurkan oleh Microsoft pada tahun 1997. Pada awalnya *game* ini dikembangkan menggunakan *game engine* Genie (sebuah *game engine* yang berbasis *sprite* 2 dimensi). Sejak awal diluncurkannya hingga sekarang, sudah terdapat beberapa perkembangan versi dari *game* ini, yaitu: *Age of Empires*, *Age of Empires II*, *Age of Empires III*, *Age of Empires Mythologies*, *Age of Empires Online*, dan yang terakhir *Age of Empires: Definitive Edition*. Antarmuka awal *game* *Age of Empires* seperti pada Gambar 2.4.

Age of Empires adalah sebuah *game* peperangan. Peperangan ini terjadi diantara beberapa kubu. *Game* ini dapat dimainkan secara *singleplayer* melawan *bot*/komputer dan dapat juga dimainkan secara *mutiplayer* hingga 4 pemain. Tujuan *game* ini adalah untuk mengalahkan raja pemain lain. Sebelum dapat melakukan penyerangan, terlebih dahulu pemain harus membangun peradaban dan pasukannya sendiri. Pemain dapat membangun berbagai macam bangunan mulai dari rumah-rumah rakyat, tempat persediaan makanan, ladang, tempat pertambangan, hingga *barack* tempat



Gambar 2.4 Antarmuka awal *game Age of Empires* [3]



Gambar 2.5 Antarmuka *gameplay game Age of Empires* [3]

pasukan. Ketika peradaban pemain sudah baik dan pasukan pemain sudah terkumpul, barulah pemain dapat melakukan penyerangan. Keseruan dari *game* ini terletak pada bagaimana setiap pemain dapat membangun peradaban dan pasukan lebih cepat dari pemain lain sehingga dapat menaklukan pasukan pemain lain dan memperluas wilayah sendiri. Antarmuka *gameplay game Age of Empires* seperti pada Gambar 2.5 [3].

2.2.3 *Chomical*

Chomical adalah sebuah *game* yang menggabungkan 2 *genre game* yang berbeda, yaitu *game* simulasi dan *game tower defense*. Penggabungan 2 *genre game* ini, belum pernah dilakukan sebelumnya. *Genre* simulasi yang diterapkan adalah percobaan kimia yang merupakan aspek kehidupan nyata yang dapat disimulasikan dalam permainan komputer sebagai bagian edukatif permainan, sementara *genre tower defense* berfungsi sebagai motivator dan aspek menyenangkan/*fun* dalam *game*. Dengan menggabungkan kedua *genre game* tersebut, tentu akan terbentuk suatu *game* edukatif mengenai simulasi percobaan kimia yang memiliki aspek *action* yang menyenangkan untuk dimainkan.



Gambar 2.6 Antarmuka awal *game Chomical* [4]

Game Chomical diimplementasikan menggunakan modul scene2D pada libGX. Modul ini dipakai untuk mengimplementasikan objek-objek pada *game* seperti tombol, *dialog*, dan tabel. Implementasi antarmuka *game Chomical* dpat dilihat pada Gambar 2.6. Terdapat beberapa hal yang harus pemain perhatikan pada saat melakukan simulasi, yaitu : takaran zat, urutan zat yang dimasukan, aturan dalam memanaskan dan mengaduk zat-zat tersebut. Jika pemain tidak memperhatikan aturan-aturan tersebut, maka pemain akan mendapatkan penalti yaitu gagalnya percobaan. Jika percobaan gagal, maka permainan akan kembali lagi ke awal percobaan.

Hasil dari simulasi yang dilakukan adalah bahan kimia yang akan digunakan untuk membuat peluru. Peluru tersebut akan digunakan pemain untuk mengalahkan musuh dalam pertempuran yang merupakan sebuah *game tower defense*. Setelah pemain mengalahkan musuh-musuh yang ada, maka pemain akan mendapatkan zat-zat baru yang dapat digunakan dalam eksperimen dalam simulasi selanjutnya [4].

2.2.4 Vector Tower Defense 2

Vector Tower Defense 2 adalah sebuah *game* yang dibuat dengan tujuan untuk mengidentifikasi proses pemain memecahkan

masalah dalam *game* dan mendeteksi potensi dari peluang pemain belajar matematika dalam *game*. Estetika pada *game* ini netral, dan tidak ada unsur kekerasan, karena target *game* ini ditujukan untuk siswa usia 10-12 tahun. Antarmuka *game Vector Tower Defense 2* dapat dilihat pada Gambar 2.7.

Dalam *game* ini, musuh harus bergerak di sepanjang jalur tertentu, dan pemain memiliki berbagai macam pilihan *tower*, *upgrade* dan *bonus point* yang dapat digunakan untuk menahan dan menghancurkan setiap musuh yang bergerak di jalur tersebut. Hal ini mendorong pemain untuk menggali keterampilan strategis mereka hingga batasnya. Salah satu fitur dari desain *game* ini adalah *bonus point* yang disertakan pada akhir setiap *wave* serangan musuh. Pemain dibatasi sejumlah uang tertentu untuk membeli *tower*, sehingga pemain harus menghitung uang yang digunakan untuk membeli *tower* tersebut, pemain pun harus menentukan waktu yang tepat untuk membeli *tower* baru atau meng-*upgrade tower* yang sudah ada. Oleh karena itu dibutuhkan aspek strategis yang baik dalam permainan [5].



Gambar 2.7 Antarmuka *game Vector Tower Defense 2* [5]

2.2.5 *Emotion*

Emotion adalah sebuah *game tower defense* yang dimainkan dengan cara memperagakan suatu emosi. Pada antarmuka *game* ini akan membutuhkan akses ke kamera yang berada pada perangkat yang dimiliki pemain. Kemudian antarmuka *game* akan menangkap gambar wajah pemain, lalu mengirim gambar wajah tersebut ke *server* untuk dilakukan penilaian dan analisis. *Dataset* emosi yang digunakan pada *game* ini adalah CIFE dan GaMo dan menggunakan model CNN dalam mengklasifikasikan emosi-emosi tersebut.

Antarmuka *game Emotion* dapat dilihat pada Gambar 2.8. Pada *game* ini, pemain akan bermain untuk melindungi sebuah desa. Pada antarmuka *game* akan muncul sebuah emosi yang harus ditirukan oleh pemain. Pada saat yang bersamaan akan muncul sebuah bom yang datang dari langit untuk menghancurkan desa. Pemain harus dapat menirukan emosi yang dimaksud sebelum bom tersebut jatuh ke desa. Jika pemain berhasil menirukan emosi yang dimaksud dengan benar, maka bom akan menghilang dan desa akan aman, sedangkan jika pemain gagal, maka bom akan jatuh sampai ke desa, kemudian meledak dan menghancurkan desa. Pemain akan dibatasi dengan *HP (Health Point)* tertentu [6].



Gambar 2.8 Antarmuka *game Emotion* [6]

2.3 Android

Android adalah sebuah sistem operasi berbasis *kernel* Linux yang awalnya dikembangkan oleh Android .Inc yang didukung Google finansial dan kemudian dibeli pada tahun 2005 hingga akhirnya diresmikan pada tahun 2007. Android dapat menyesuaikan spesifikasi di kelas *low-end* hingga *high-end*. Android memiliki nama-nama unik disetiap versinya yang diambil dari nama-nama makanan dan diurutkan berdasarkan huruf alfabet [7].

2.4 Game Engine Unity

Unity adalah sebuah *game engine* yang dikembangkan oleh Unity Technologies. Unity pertama kali diluncurkan pada tahun 2005 dan terus mengalami perkembangan sampai dengan versi terbarunya yaitu versi 2018. Unity ini mendukung untuk pembuatan *game* berbasis 2 dimensi maupun 3 dimensi. Selain itu, Unity pun mampu mengambil aset gambar dari berbagai perangkat lunak. *Game engine* ini pun mendukung beberapa bahasa pengembangan aplikasi seperti C#, *UnityScript* (berbentuk *JavaScript*), dan *BooScript* yang dapat terintegrasi dengan bahasa *Python*. Hasil *game* yang telah dibuat dengan menggunakan Unity ini nantinya dapat mendukung berbagai macam *platform game* seperti PC, mobile (Android, iOS, Windows Phone), *web*, dan bahkan *console gaming* [8].

(Halaman Ini Sengaja dikosongkan)

BAB III

ANALISIS DAN PERANCANGAN

Pada bab ini akan dibahas mengenai analisis dan perancangan yang akan digunakan untuk menyelesaikan tugas akhir. Bab ini terdiri dari analisis sistem, perancangan *game*, dan perancangan antarmuka.

3.1 Analisis Sistem

Pada saat ini sudah terdapat berbagai macam *game* dengan *genre tower defense*. Pada usulan tugas akhir ini, *game* yang akan dibuat adalah *game* dengan *genre tower defense* yang akan dimainkan melawan *bot/komputer*. *Bot* akan dirancang sedemikian rupa agar memiliki tingkat kesulitan yang meningkat seiring bertambahnya *level*. Cara bermain yang digunakan adalah dengan memilih pasukan untuk bertahan dari serangan musuh sekaligus menghancurkan wilayah musuh untuk memenangkan permainan. Dengan sistem seperti ini, akan membuat permainan menjadi lebih strategis dan menarik.

Dalam membangun *game* ini, penulis menggunakan *game engine* Unity versi 2017.2.0f3 dengan bahasa pemrograman C#. Untuk antarmuka *game* didesain menggunakan aplikasi Corel Draw X7, Adobe Illustrator CS6, dan Adobe Photoshop CS6. *Game* ini dapat berjalan di *smartphone* berbasis Android, sehingga *game* akan menjadi semakin menyenangkan dan mudah untuk dimainkan dimana saja dan kapan saja.

3.2 Perancangan Game

Subbab ini membahas perancangan *game* yang akan digunakan untuk tugas akhir. Perancangan *game* yang dibahas adalah deskripsi umum *game*, aturan permainan yang meliputi cara bermain dan kondisi menang atau kalah, skenario yang terdiri dari skenario *hero* dan skenario *monster*, serta tingkat kesulitan pada setiap *level*.

3.2.1 Deskripsi Umum *Game*

Game The Legend Guardians merupakan *game* yang memiliki *genre Tower Defense*. *Game* ini dapat dimainkan pada *platform* Android. Seperti pada *game* dengan *genre Tower Defense* yang lainnya, *game The Legend Guardians* ini dimainkan dengan cara memilih pasukan yang ada untuk mempertahankan wilayah kekuasaan pemain dari serangan musuh sekaligus menyerang untuk menghancurkan wilayah kekuasaan musuh. Terdapat berbagai macam pasukan yang memiliki kemampuan yang berbeda-beda. Perbedaan setiap pasukan tersebut terdapat pada *cost* (harga), *HP (Health Point)*, *power* (kekuatan), dan *attack speed* (kecepatan menyerang), *movement speed* (kecepatan bergerak), dan *cooldown* (jeda untuk keluar lagi).

Pada *game* ini, terdapat kubu-kubu yang memiliki wilayah masing-masing untuk dipertahankan. Pemain dapat memilih berbagai jenis pasukan yang memiliki kemampuan yang berbeda-beda, diantaranya: *saber* (memiliki kekuatan, pertahanan, dan kecepatan yang seimbang), *archer* (menyerang dari jarak jauh), *lancer* (memiliki kekuatan, pertahanan, dan kecepatan menengah), *assasin* (menyerang dengan cepat namun pertahanannya rendah), *caster* (menyerang dari jarak jauh menggunakan sihir), *berserker* (memiliki pertahanan yang besar namun kecepatannya rendah), dan *rider* (memiliki kekuatan, pertahanan, dan kecepatan yang tinggi). Untuk mengeluarkan pasukan, pemain dibatasi oleh uang yang akan berkurang sesuai harga dari pasukan tersebut. Uang ini akan bertambah kembali dalam jangka waktu tertentu. Setiap pasukan pun memiliki jeda waktu (*cooldown*) setiap kali dikeluarkan.

Game ini dimainkan dengan cara *singleplayer*, pemain akan berpetualang untuk mengalahkan *monster* yang berbeda-beda di setiap *level*. Pemain akan bermain melawan *bot* (komputer). *Bot* akan dirancang sedemikian rupa agar memiliki tingkat kesulitan yang meningkat seiring bertambahnya *level*. Cara bermain yang digunakan adalah dengan memilih pasukan untuk bertahan dari serangan musuh sekaligus menghancurkan wilayah musuh untuk

memenangkan permainan. Dengan sistem seperti ini, akan membuat permainan menjadi lebih strategis dan menarik.

3.2.2 Aturan Permainan

Aturan permainan dalam *game* ini adalah sebagai berikut :

- Cara bermain *game* ini adalah dengan cara menekan *icon-icon* pasukan yang akan dikeluarkan ke medan peertempuran, *icon-icon* ini nantinya disebut sebagai *hero buttons*. Setiap pasukan memiliki kekuatan dan harga yang berbeda-beda.
- Setiap pasukan yang dikeluarkan akan berjalan menuju basis pertahanan musuh. Jika mereka berhadapan dengan musuh yang menghalangi jalan mereka, maka mereka akan saling bertempur.
- Seiring berjalannya waktu, pemain menghasilkan uang yang akan digunakan untuk membeli pasukan yang akan dikirim ke medan perang.
- Setiap 1 detik pemain akan mendapatkan uang 10 *gold*.
- Setiap membunuh *monster* pemain akan mendapatkan uang dengan jumlah berbeda-beda. Untuk *monster Blue, Purple, Green*, dan *Yellow* akan mendapatkan 10 *gold*, sedangkan untuk *monster Orange* dan *Cream* akan mendapatkan 20 *gold*, dan untuk *monster Red* akan mendapatkan 30 *gold*.
- Maksimum uang : 200 *gold*.
- Kondisi menang : ketika pemain dapat bertahan dan basis pertahanan musuh hancur yaitu ketika *HP (Health Point) Tower* musuh mencapai 0.
- Kondisi kalah : ketika basis pertahanan milik sendiri hancur yaitu ketika *HP (Health Point) Tower* mencapai 0.

3.2.3 Skenario

Terdapat 2 skenario yang dibuat dalam *game* ini, yaitu skenario *hero* dan skenario *monster*. Setiap *hero* dan *monster* memiliki kemampuan yang berbeda-beda. Skenario *hero* dapat dilihat pada Tabel 3.1 sedangkan skenario *monster* dapat dilihat pada Tabel 3.2.

Tabel 3.1 Skenario Hero

<i>Hero</i>	<i>Cost</i>	<i>HP</i>	<i>Power</i>	<i>Movement Speed</i>	<i>Attack Speed</i>	<i>Cooldown (second)</i>
<i>Saber</i>	30 gold	100	25	20	10	3
<i>Archer</i>	30 gold	100	25	15	10	3
<i>Lancer</i>	50 gold	200	35	20	15	5
<i>Assasin</i>	50 gold	150	35	25	20	5
<i>Caster</i>	60 gold	150	50	15	7.5	6
<i>Berserker</i>	80 gold	300	75	20	5	8
<i>Rider</i>	100 gold	350	80	30	15	10

Tabel 3.2 Skenario Monster

<i>Monster</i>	<i>HP</i>	<i>Power</i>	<i>Movement Speed</i>	<i>Attack Speed</i>
<i>Blue</i>	100	25	20	10
<i>Purple</i>	100	25	15	10
<i>Green</i>	150	35	25	15
<i>Yellow</i>	150	40	20	10
<i>Orange</i>	200	60	25	15
<i>Cream</i>	200	60	15	10
<i>Red</i>	350	80	20	5

3.2.4 Tingkat Kesulitan

Seperti halnya dalam berbagai macam *game* lainnya, pada *game* ini pun memiliki tingkat kesulitan yang beragam sesuai *level* nya. Semakin tinggi *level*, umumnya tingkat kesulitannya akan semakin meningkat. Tingkat kesulitan pada *game* ini dibuat sedemikian rupa agar membuat pemain tertantang dan merasa ingin terus bermain hingga dapat menyelesaikan semua *level*. Tingkat kesulitan pada *game* ini dapat dilihat pada Tabel 3.3.

Tabel 3.3 Tingkat Kesulitan

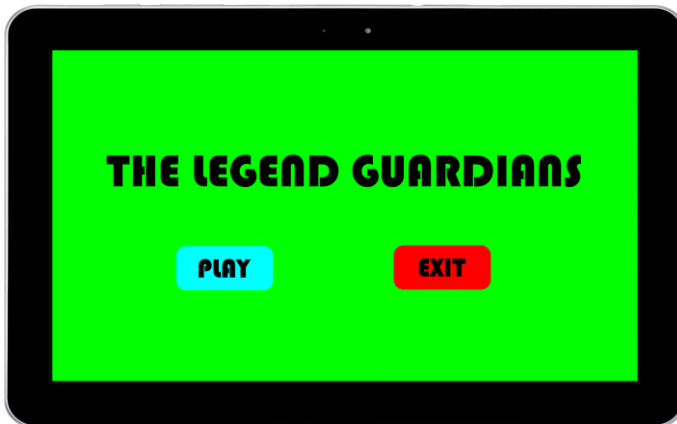
<i>Level</i>	<i>Starting Gold</i>	<i>HP Tower Player</i>	<i>HP Tower Enemy</i>
1	0	100	100
2	0	100	150
3	30	150	200
4	20	150	250
5	50	200	300
6	40	250	400
7	70	250	350
8	60	250	400
9	50	300	450
10	100	300	500
11	90	350	550
12	80	450	750
13	120	350	600
14	110	400	650
15	100	400	700
16	150	450	750
17	130	450	800
18	150	500	1000

3.3 Perancangan Antarmuka

Subbab ini membahas bagaimana rancangan antarmuka pengguna yang akan digunakan untuk tugas akhir. Rancangan antarmuka yang dibahas dalam *game* ini terdapat beberapa antarmuka, yaitu antarmuka awal, antarmuka menu utama, antarmuka *tutorial*, antarmuka *select level*, dan antarmuka *game*.

3.3.1 Rancangan Antarmuka Awal

Antarmuka awal adalah antarmuka yang pertama kali muncul ketika *game* dijalankan. Pada antarmuka awal, terdapat dua hal yang dapat dilakukan pemain, yaitu masuk ke menu utama dengan menekan tombol *Play* atau keluar dari *game* dengan menekan tombol *Exit*. Rancangan antarmuka awal dapat dilihat pada Gambar 3.1.



Gambar 3.1 Rancangan antarmuka awal

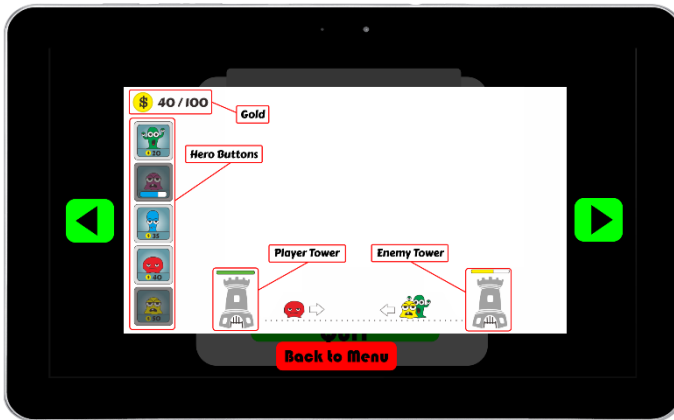
3.3.2 Rancangan Antarmuka Menu Utama

Antarmuka menu utama adalah antarmuka yang akan muncul setelah pemain menekan tombol *Play* pada antarmuka awal. Rancangan antarmuka menu utama dapat dilihat pada Gambar 3.2. Pada antarmuka menu utama, terdapat beberapa tombol yang dapat dipilih oleh pemain, yaitu :

1. Tombol *Start* akan digunakan pemain untuk memulai permainan dan masuk ke antarmuka *select level*.
2. Tombol *Tutorial* akan digunakan pemain untuk mengetahui beberapa hal tentang *game* ini termasuk penjelasan bagaimana cara bermain *game* ini, hingga kondisi menang dan kondisi kalah.
3. Tombol *Credit* akan digunakan pemain untuk mengetahui *credit* dari *game* yang terdiri dari *developer*, *designer*, dan *source* dari musik maupun *sound effect* yang digunakan dalam *game*.
4. Tombol *Quit* akan digunakan pemain untuk keluar dan kembali ke antarmuka awal.



Gambar 3.2 Rancangan antarmuka menu utama



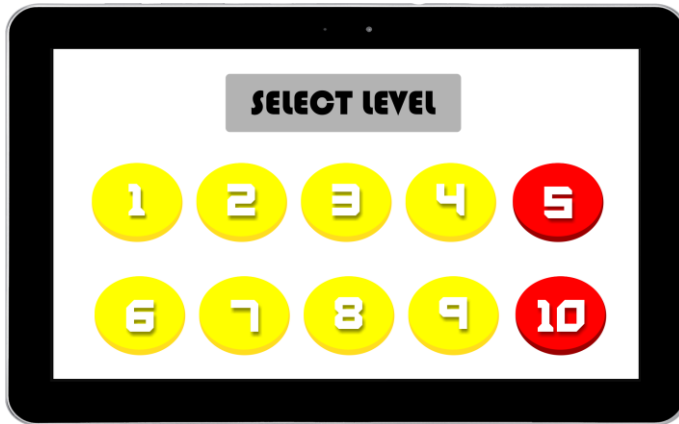
Gambar 3.3 Rancangan antarmuka *tutorial*

3.3.3 Rancangan Antarmuka *Tutorial*

Antarmuka *tutorial* adalah antarmuka yang akan muncul setelah pemain menekan tombol *Tutorial* pada antarmuka menu utama. Penjelasan yang disajikan pada antarmuka *tutorial* adalah *slide* yang berisi gambar-gambar yang menjelaskan bagaimana cara bermain *game* ini, hingga kondisi menang dan kondisi kalah. Rancangan antarmuka *tutorial* dapat dilihat pada Gambar 3.3.

3.3.4 Rancangan Antarmuka *Select Level*

Antarmuka *select level* adalah antarmuka yang akan tampil ketika pemain menekan tombol *Start* pada antarmuka menu utama. Pada antarmuka ini akan ditampilkan beberapa pilihan *level* yang sudah terbuka. Kemudian pemain dapat memilih *level* yang ingin diselesaikan. Jika pemain berhasil menyelesaikan suatu *level*, maka *level* selanjutnya akan terbuka. Rancangan antarmuka *select level* dapat dilihat pada Gambar 3.4.

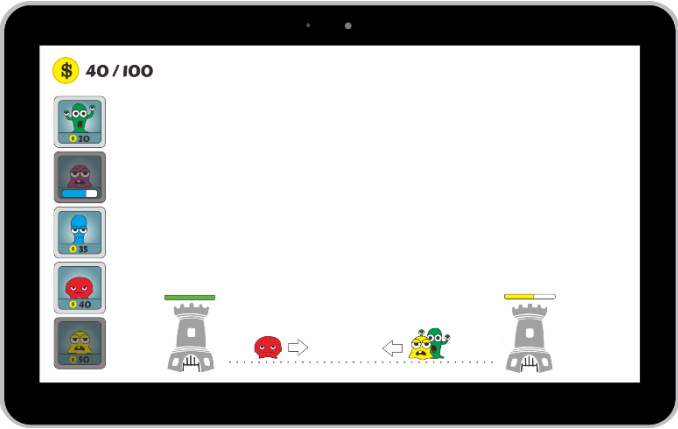


Gambar 3.4 Rancangan antarmuka *select level*

3.3.5 Rancangan Antarmuka *Game*

Antarmuka *game* adalah antarmuka yang akan muncul ketika pemain sudah memilih suatu *level*. Rancangan antarmuka *game* dapat dilihat pada Gambar 3.5. Berikut ini adalah penjelasan dari bagian-bagian yang terdapat pada antarmuka *game* :

1. *Gold* yang digunakan untuk membeli pasukan.
2. *Hero Buttons* adalah menu pilihan pasukan.
3. *Tower* sebagai basis pertahanan pemain.
4. *Tower Enemy* sebagai basis pertahanan musuh.
5. Jalur yang dilalui pasukan untuk menyerang basis pertahanan musuh.



Gambar 3.5 Rancangan antarmuka *game*

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan *game* yang dibuat. Bab ini mencakup proses penerapan dan pengimplementasian dalam bentuk *code*, dan antarmuka yang mengacu pada rancangan yang telah dibahas sebelumnya.

4.1 Lingkungan Implementasi

Lingkungan implementasi dari tugas akhir ini adalah perangkat *smartphone* yang berbasis Android. Selain itu, *game* ini juga dapat dimainkan di komputer.

4.2 Implementasi Permainan

Implementasi dari masing-masing fungsi utama dituliskan menggunakan *code* berbahasa C# pada *game engine* Unity. Implementasi fungsi diurut berdasarkan antarmuka-antarmuka yang ada pada *game*.

4.2.1 Implementasi Antarmuka Awal

Antarmuka awal adalah antarmuka yang pertama kali muncul ketika *game* dijalankan. Antarmuka awal dapat diimplementasikan sesuai rancangannya dengan memperbaiki tampilannya agar menjadi lebih menarik. Pada antarmuka ini, terdapat dua hal yang dapat dilakukan pemain, yaitu masuk ke menu utama dengan menekan tombol *Play* atau keluar dari *game* dengan menekan tombol *Exit*. Implementasi antarmuka awal dapat dilihat pada Gambar 4.1.



Gambar 4.1 Implementasi antarmuka awal

Fungsi yang digunakan untuk menjalankan antarmuka awal dapat dilihat pada Kode Sumber 4.1, berikut ini adalah penjelasannya :

1. *SceneManager.LoadScene("MainMenu")*
Fungsi ini dijalankan setelah pemain menekan tombol *Play*. Dengan menggunakan fungsi ini, sistem akan membaca dan berpindah ke *scene* yang ingin dituju, yaitu *scene "MainMenu"*.
2. *Application.Quit()*
Fungsi ini digunakan untuk keluar dari *game*.

```
using UnityEngine.SceneManagement;
public void Play(){
    SceneManager.LoadScene("MainMenu");
}

public void Exit(){
    Application.Quit();
}
```

Kode Sumber 4.1 Fungsi antarmuka awal

4.2.2 Implementasi Antarmuka Menu Utama



Gambar 4.2 Implementasi antarmuka menu utama

Antarmuka menu utama adalah antarmuka yang akan muncul setelah pemain menekan tombol *Play* pada antarmuka awal. Antarmuka menu utama dapat diimplementasikan sesuai rancangannya. Implementasi antarmuka menu utama dapat dilihat pada Gambar 4.2. Pada antarmuka menu utama, terdapat beberapa tombol yang dapat dipilih oleh pemain, yaitu :

1. Tombol *Start*. Ketika pemain memilih tombol *Start*, pemain akan memulai permainan melawan *bot*. Namun sebelum itu, pemain akan masuk ke antarmuka *select level* terlebih dahulu untuk memilih *level* yang akan dimainkannya.
2. Tombol *Tutorial*. Ketika pemain memilih tombol *Tutorial*, akan muncul beberapa *slide* gambar yang menjelaskan tentang *game* ini. Terdapat juga penjelasan bagaimana cara bermain *game* ini, hingga kondisi menang, dan kondisi kalah.
3. Tombol *Credit*. Ketika pemain memilih tombol *Credit*, akan muncul *credit* dari *game* yang dibuat. *Credit* ini terdiri dari

developer, designer, dan source dari musik maupun *sound effect* yang digunakan dalam *game*.

4. Tombol *Quit*. Ketika pemain memilih tombol *Quit*, maka akan muncul menu konfirmasi untuk keluar dari *game* atau tidak, jika pemain mengkonfirmasi untuk keluar, maka pemain akan keluar dari *game* dan jika pemain tidak mengkonfirmasi untuk keluar, maka menu konfirmasi akan tertutup dan pemain akan kembali ke antarmuka menu utama.

Fungsi-fungsi yang digunakan pada implementasi antarmuka menu utama adalah fungsi-fungsi untuk menampilkan atau menutup antarmuka setiap panel dan untuk *load scene* yang lainnya. Fungsi-fungsi menu utama tersebut dapat dilihat pada Kode Sumber 4.2.

```
public void StartGame() {
    StartPanel.SetActive(true);
}

public void StartSingleplayer() {
    SceneManager.LoadScene("SelectLevel");
}

public void Tutorial() {
    TutorialPanel.SetActive(true);
}

public void Credit() {
    CreditPanel.SetActive(true);
}

public void QuitGame() {
    ConfirmQuitPanel.SetActive(true);
}

public void Back() {
    StartPanel.SetActive(false);
    TutorialPanel.SetActive(false);
}
```

```

        CreditPanel.SetActive(false);
        ConfirmQuitPanel.SetActive(false);
    }

    public void QuitYes() {
        Application.Quit();
    }
}

```

Kode Sumber 4.2 Fungsi antarmuka menu utama

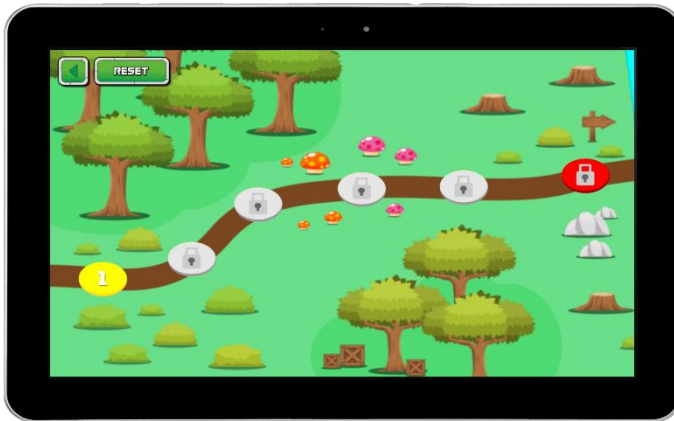
4.2.3 Implementasi Antarmuka *Tutorial*

Antarmuka *tutorial* adalah antarmuka yang akan muncul setelah pemain menekan tombol *Tutorial* pada antarmuka menu utama. Antarmuka *tutorial* dapat diimplementasikan sesuai rancangannya. Implementasi antarmuka *tutorial* dapat dilihat pada Gambar 4.3.



Gambar 4.3 Implementasi antarmuka *tutorial*

4.2.4 Implementasi Antarmuka *Select Level*



Gambar 4.4 Implementasi antarmuka *select level*

Antarmuka *select level* adalah antarmuka yang akan muncul ketika pemain memilih menu *Start* pada antarmuka menu utama. Antarmuka *select level* dapat diimplementasikan sesuai rancangannya. Bahkan dibuat menjadi bentuk peta agar tampilannya menjadi lebih menarik. Pemain dapat menggeser layar untuk dapat memilih pilihan *level* yang lebih banyak. Implementasi antarmuka *select level* dapat dilihat pada Gambar 4.4. Pada implementasi antarmuka ini terdapat beberapa bagian penting, yaitu :

1. *Level* yang sudah terbuka.
2. *Level* yang belum terbuka. Untuk membuka *level* ini, pemain harus berhasil menyelesaikan *level* sebelumnya.
3. Tombol *Reset* untuk mengulang progress permainan.
4. Tombol *Back* untuk kembali ke antarmuka menu utama.

Fungsi yang digunakan pada implementasi antarmuka *select level* adalah fungsi untuk menandai setiap *button level*


```

public static int Level;
public int max_Level;
public GameObject[] LevelUnlocked;

void Start () {
    Level = PlayerPrefs.GetInt("Level", Level);
}

void Update() {
    for (int i = 1; i < max_Level; i++) {
        if (i <= Level)
            LevelUnlocked[i].SetActive(false);
        else
            LevelUnlocked[i].SetActive(true);
    }
}

public static void Next_Level() {
    if (Level == NextLevel.theLevel) {
        Level += 1;
        PlayerPrefs.SetInt("Level", Level);
    }
}

```

Kode Sumber 4.3 Fungsi antarmuka *select level*

dengan penanda *level* tertentu, jika suatu *level* berhasil diselesaikan, maka penanda *level* berikutnya akan diaktifkan agar *level* tersebut dapat dimainkan. Fungsi *select level* tersebut dapat dilihat pada Kode Sumber 4.3.

4.2.5 Implementasi Antarmuka *Game*

Antarmuka *game* adalah antarmuka yang akan muncul ketika pemain sudah memilih suatu *level* pada antarmuka *select level*. Antarmuka *game* dapat diimplementasikan sesuai rancangannya. Implementasi antarmuka *game* dapat dilihat pada Gambar 4.5.



Gambar 4.5 Implementasi antarmuka *game*

Pada implementasi antarmuka *game* terdapat bagian-bagian yang penting, yaitu:

1. Keterangan *level* digunakan sebagai penanda *level* yang sedang dimainkan.
2. *Gold* digunakan sebagai uang untuk membeli pasukan. *Gold* akan bertambah seiring berjalannya waktu dan ketika pemain membunuh *monster*.
3. Tombol *Pause* digunakan untuk memberhentikan sejenak *game*. Pada saat *pause*, suatu menu akan muncul, pemain dapat melanjutkan kembali permainan atau kembali ke antarmuka *select level*.
4. *Hero buttons* digunakan sebagai menu pilihan pasukan. Terdapat berbagai macam pasukan yang dapat dibeli dan dikirim ke medan pertempuran, setiap pasukan pun memiliki harga dan kemampuan yang berbeda-beda.
5. *Tower* yang digunakan sebagai basis pertahanan pemain yang harus dijaga.

6. *Tower enemy* digunakan sebagai basis pertahanan musuh yang harus dihancurkan.

Terdapat beberapa fungsi penting yang digunakan dalam implementasi antarmuka *game*, yaitu :

1. *CurrencyChange()*

Fungsi ini digunakan untuk menambahkan uang pemain sebesar 10 *gold* setiap detik. Uang pemain dapat bertambah ketika pemain membunuh *monster* musuh. *Monster* musuh yang dibunuh akan menghasilkan uang yang berbeda-beda. Uang pemain pun dapat berkurang ketika membeli pasukan sesuai dengan harga pasukan tersebut karena setiap pasukan memiliki harga yang berbeda-beda.

2. *BuyHero (HeroButton heroButton)*

Fungsi ini digunakan pada saat pemain membeli pasukan dengan cara menekan *hero buttons*. Jika pemain memiliki *gold* yang cukup atau lebih dari harga pasukan tersebut, maka fungsi ini akan dijalankan.

3. *SpawnHero()*

Fungsi ini digunakan untuk mengeluarkan pasukan sesuai *flag* dari *hero buttons* yang ditekan.

4. *CheckGameOver()*

Fungsi ini digunakan untuk mengecek keadaan *game*. Hal yang dicek pada fungsi ini adalah keadaan *tower* pemain dan *tower* musuh. Apabila salah satunya telah hancur, maka permainan akan berakhir. Jika *tower* pemain hancur, maka tampilan *lose game* akan muncul. Sedangkan jika *tower* musuh hancur, maka tampilan *win game* akan muncul.

Fungsi-fungsi tersebut dapat dilihat pada Kode Sumber 4.4. Pada implementasi *game* terdapat 18 *level*. Semakin tinggi *level* maka tingkat kesulitannya akan semakin meningkat. Setiap *level* memiliki berbagai macam perbedaan, diantaranya :

1. *Starting gold* atau uang awal.
2. *HP (Health Point)* tower pemain
3. *HP* tower musuh.
4. Jenis *hero* / pasukan yang dapat dikeluarkan
5. Jenis dan jumlah *monster* yang akan muncul.
6. *Spawning type* pada suatu *level*.
7. Batas maksimum waktu suatu *level*.

```

void Update() {
    CurrencyChange();
}

public void CurrencyChange() {
    int nextTime += Time.deltaTime;
    if (Currency < maxGold) {
        if (nextTime >= interval) {
            Currency += 10;
            nextTime = 0;
        }
    }
}

public void BuyHero(HeroButton heroButton) {
    flagHero = heroButton.flag;
    if (Currency >= heroButton.Price)
        Currency -= heroButton.Price;
    SpawnHero();
}

public void SpawnHero() {
    flag = flagHero;
    if (flag == 1)
        Hero Hero1 = Instantiate(Hero1);
    else if (flag == 2)
        Hero Hero2 = Instantiate(Hero2);
    else if (flag == 3)
        Hero Hero2 = Instantiate(Hero3);
    else if (flag == 4)
        Hero Hero2 = Instantiate(Hero4);
}

```

```

else if (flag == 5)
    Hero Hero2 = Instantiate(Hero5);
else if (flag == 6)
    Hero Hero2 = Instantiate(Hero6);
else if (flag == 7)
    Hero Hero2 = Instantiate(Hero7);
}

public void checkGameOver() {
    if (liveTower == 0 )
        LoseCanvas.SetActive(true);
    else if (liveTowerEnemy == 0)
        WinCanvas.SetActive(true);
}

```

Kode Sumber 4.4 Fungsi antarmuka *game*

Selain fungsi-fungsi tersebut, terdapat juga kode sumber yang berisikan fungsi-fungsi yang digunakan untuk *hero*, *monster*, dan *tower*. Fungsi-fungsi tersebut adalah :

1. *Move()*
Fungsi ini digunakan ketika *hero* dan *monster* bergerak.
2. *Attack()*
Fungsi ini digunakan ketika *hero* dan *monster* menyerang.
3. *TakeDamage()*
Fungsi ini digunakan ketika *hero*, *monster*, dan *tower* menerima serangan.
4. *isAlive()*
Fungsi ini digunakan untuk mengecek apakah *hero*, *monster*, dan *tower* masih hidup.
5. *Die()*
Fungsi ini digunakan ketika *hero*, *monster*, dan *tower* mati.

```
void Start () {
    health = startHP;
    healthBar.fillAmount = 1;
    myAnimator = gameObject.GetComponent<Animator>();
}

void Update () {
    Move();
    Attack();
}

void OnTriggerEnter2D(Collider2D other) {
    if (other.tag == "TowerMonster") {
        canAttack = 1;
        speed = 0;
        _target = 2;
        monsterCount++;
    }

    else if (other.tag == "Monster") {
        canAttack = 1;
        speed = 0;
        _target = 1;
        monsterCount ++;
    }
}

void OnTriggerExit2D(Collider2D other) {
    if (other.tag == "Monster") {
        monsterCount --;
        if (monsterCount == 0)
            canAttack = 0;
    }

    else if (other.tag == "TowerMonster") {
        monsterCount --;
        StartCoroutine(idleAnimation());
    }
}
```

```

public void TakeDamage(float damage) {
    health -= damage;
    healthBar.fillAmount = health / startHP;
    IsAlive();
}

public void IsAlive() {
    if (health <= 0) {
        //Die();
        speed = 0;
        Destroy(gameObject);
    }

    else {
        speed = startSpeed;
        GetComponent<BoxCollider2D>().enabled = true;
    }
}

private void Move() {
    if (attack == 0 && canAttack == 0 && monsterCount
        == 0 && die == false) {
        transform.position += direction * speed *
            Time.deltaTime;
        speed = startSpeed;
        StartCoroutine(moveAnimation());
    }
}

private void Attack() {
    if (attack == 0 && canAttack == 1 && monsterCount
        != 0 && die == false) {
        speed = 0;
        StartCoroutine(idleAnimation());
        attackTimer += Time.deltaTime;

        if (attackTimer >= attackCooldown && die ==
            false) {
            attack = 1;
            attackTimer = 0;
        }
    }
}

```

```

        }

        if (attack == 1) {
            StartCoroutine(attackAnimation());
        }
    }

private void Die() {
    die = true;
    attack = 2; canAttack = 2;
    speed = 0;
    GetComponent<BoxCollider2D>().enabled = false;

    StartCoroutine(dieAnimation());

    GetComponent<SpriteRenderer>().sortingOrder = 0;
    healthBar.canvas.sortingOrder = -5;
    Destroy(gameObject);
}

```

Kode Sumber 4.5 Fungsi pada *hero*

Fungsi-fungsi yang digunakan pada *hero* dan *monster* adalah fungsi-fungsi yang sama. Perbedaan nya terdapat pada target dari *other.tag* nya. *Hero* memiliki target *monster* dan *tower monster*, sebaliknya *monster* memiliki *hero* dan *tower hero*. Kemudian fungsi-fungsi yang digunakan pada *tower* juga sama, hanya *tower* tidak memiliki fungsi *Move()* dan *Attack()*.

BAB V

PENGUJIAN DAN EVALUASI

Pada bab ini akan dijelaskan mengenai rangkaian uji coba dan evaluasi yang dilakukan. Proses pengujian dilakukan menggunakan metode *blackbox* untuk menguji setiap fungsionalitas pada sistem dengan skenario yang telah ditentukan [9], pengujian alur permainan, pengujian tingkat kesulitan, dan pengujian *usability* yang dilakukan dengan survei.

5.1 Lingkungan Uji Coba

Lingkungan pengujian sistem pada pengerjaan Tugas Akhir ini dilakukan pada lingkungan dan alat kakas sebagai berikut:

Jenis Perangkat : Asus Zenfone Selfie Z00UD
 Sistem Operasi : Android 5.0 Lollipop
 Chipset : Qualcomm MSM8939 Snapdragon 615
 CPU : Octa-core (4x1.7 GHz Cortex-A53 & 4x1.0 GHz Cortex-A53)
 GPU : Adreno 405
 RAM : 3 GB

5.2 Pengujian Fungsionalitas

Pengujian fungsionalitas sistem dilakukan dengan menyiapkan sejumlah skenario sebagai tolok ukur keberhasilan pengujian. Pengujian fungsionalitas dilakukan dengan mengacu pada rancangan antarmuka yang telah dijelaskan pada subbab 3.3.

Pada subbab ini dijelaskan secara detil mengenai skenario yang dilakukan dan hasil yang didapatkan dari pengujian fungsionalitas *game* yang dibangun. Penjelasan disajikan dengan menampilkan kondisi awal, masukan, keluaran yang diharapkan, hasil uji coba, dan kondisi akhir. Pengujian fungsionalitas yang terdapat pada *game* ini dijabarkan sebagai berikut:

5.2.1 Uji Coba Pada Antarmuka Awal

Uji coba fungsionalitas yang akan dilakukan pada antarmuka awal adalah skenario yang terjadi ketika pemain menekan tombol *Start* dan tombol *Exit*. Hasil uji coba antarmuka awal dapat dilihat pada Tabel 5.1.

Tabel 5.1 Hasil uji coba antarmuka awal

ID	UF-001
Nama	Uji Coba Pada Antarmuka Awal
Tujuan uji coba	Pemain mengetahui fungsionalitas tombol-tombol antarmuka awal
Kondisi awal	Pemain berada pada antarmuka awal, terdapat suara latar belakang
<i>Skenario 1</i>	<i>Pemain menekan tombol Start</i>
Masukan	Klik tombol <i>Start</i>
Keluaran yang diharapkan	Pemain berpindah ke antarmuka menu utama
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada di antarmuka menu utama
<i>Skenario 2</i>	<i>Pemain menekan tombol Exit</i>
Masukan	Klik tombol <i>Exit</i>
Keluaran yang diharapkan	Tampilan <i>game</i> ditutup
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain keluar dari <i>game</i>

5.2.2 Uji Coba Pada Antarmuka Menu Utama

Uji coba fungsionalitas yang akan dilakukan pada antarmuka menu utama adalah skenario yang terjadi ketika menekan tombol *Play*, tombol *Tutorial*, tombol *Credit*, dan tombol *Quit*. Hasil uji coba antarmuka menu utama dapat dilihat pada Tabel 5.2.

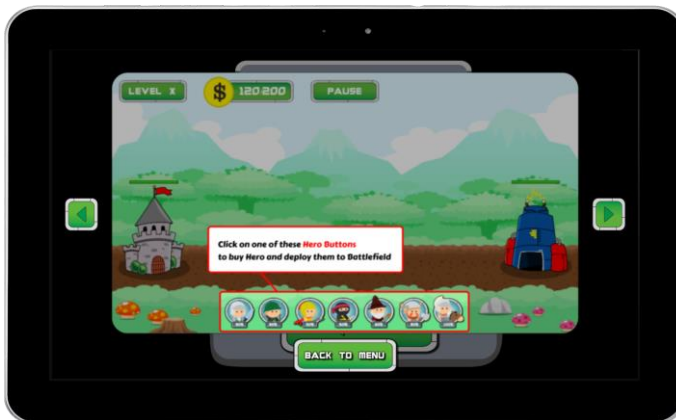
Tabel 5.2 Hasil uji coba antarmuka menu utama

ID	UF-002
Nama	Uji Coba Pada Antarmuka Menu Utama
Tujuan uji coba	Pemain mengetahui fungsionalitas tombol-tombol pada antarmuka menu utama
Kondisi awal	Pemain berada pada antarmuka menu utama, terdapat suara latar belakang
<i>Skenario 1</i>	<i>Pemain menekan tombol Start</i>
Masukan	Klik tombol Start
Keluaran yang diharapkan	Pemain berpindah ke antarmuka <i>select level</i>
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada di antarmuka <i>select level</i>
<i>Skenario 2</i>	<i>Pemain menekan tombol Tutorial</i>
Masukan	Klik tombol <i>Tutorial</i>
Keluaran yang diharapkan	Menampilkan <i>slide tutorial</i>
Hasil uji coba	Berhasil
Kondisi Akhir	<i>Slide tutorial</i> tampil
<i>Skenario 3</i>	<i>Pemain menekan tombol Credit</i>
Masukan	Klik tombol Credit
Keluaran yang diharapkan	Menampilkan <i>Credit</i> dari <i>game</i>
Hasil uji coba	Berhasil

Kondisi Akhir	<i>Credit dari game tampil</i>
<i>Skenario 4</i>	<i>Pemain menekan tombol Quit</i>
Masukan	Klik tombol <i>Quit</i>
Keluaran yang diharapkan	Tampilan <i>game</i> ditutup
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain keluar dari <i>game</i>

5.2.3 Uji Coba Pada Antarmuka *Tutorial*

Uji coba fungsionalitas yang akan dilakukan pada antarmuka *tutorial* adalah skenario yang terjadi ketika menekan tombol *Next*, tombol *Previous*, dan tombol *Back to Menu*. Uji coba pada antarmuka ini dilakukan dengan skenario *slide* no 2 dalam keadaan aktif. Skenario uji coba tersebut dapat dilihat pada Gambar 5.1. Hasil uji coba antarmuka *tutorial* dapat dilihat pada Tabel 5.3.



Gambar 5.1 Uji coba antarmuka *tutorial*

Tabel 5.3 Hasil uji coba antarmuka *tutorial*

ID	UF-003
Nama	Uji Coba Pada Antarmuka <i>Tutorial</i>
Tujuan uji coba	Pemain mengetahui fungsionalitas tombol-tombol pada antarmuka <i>tutorial</i>
Kondisi awal	Pemain berada pada antarmuka <i>tutorial</i> . <i>Slide</i> nomor 2 sedang ditampilkan
<i>Skenario 1</i>	<i>Pemain menekan tombol Next</i>
Masukan	Klik tombol <i>Next</i>
Keluaran yang diharapkan	Menampilkan <i>Slide</i> nomor 3
Hasil uji coba	Berhasil
Kondisi Akhir	<i>Slide</i> nomor 3 ditampilkan
<i>Skenario 2</i>	<i>Pemain menekan tombol Previous</i>
Masukan	Klik tombol <i>Previous</i>
Keluaran yang diharapkan	Menampilkan <i>Slide</i> nomor 1
Hasil uji coba	Berhasil
Kondisi Akhir	<i>Slide</i> nomor 1 ditampilkan
<i>Skenario 3</i>	<i>Pemain menekan tombol Back to Menu</i>
Masukan	Klik tombol <i>Back to Menu</i>
Keluaran yang diharapkan	Pemain berpindah ke antarmuka menu utama
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada di antarmuka menu utama

5.2.4 Uji Coba Pada Antarmuka *Select Level*

Uji coba fungsionalitas yang akan dilakukan pada antarmuka *select level* adalah skenario yang terjadi ketika menekan tombol-tombol *Level*, tombol *Reset*, dan tombol *Back*. Uji coba pada antarmuka ini dilakukan dengan skenario *level* 1 hingga 3

terbuka sedangkan *level 4* dan seterusnya tertutup. Skenario uji coba tersebut dapat dilihat pada Gambar 5.2. Hasil uji coba antarmuka *select level* dapat dilihat pada Tabel 5.4.



Gambar 5.2 Skenario uji coba antarmuka *select level*

Tabel 5.4 Hasil uji coba antarmuka *select level*

ID	UF-004
Nama	Uji Coba Pada Antarmuka <i>Select Level</i>
Tujuan uji coba	Pemain mengetahui fungsionalitas tombol-tombol pada antarmuka <i>select level</i>
Kondisi awal	Pemain berada pada antarmuka <i>select level</i> , <i>level 1</i> hingga <i>3</i> terbuka sedangkan <i>level 4</i> dan seterusnya tertutup, terdapat suara latar belakang
<i>Skenario 1</i>	<i>Pemain menekan tombol level 1</i>
Masukan	Klik tombol <i>level 1</i>
Keluaran yang diharapkan	Pemain berpindah ke antarmuka <i>level 1</i> dan memulai permainan
Hasil uji coba	Berhasil

Kondisi Akhir	Pemain berada di antarmuka <i>level 1</i> dan permainan dimulai
<i>Skenario 2</i>	<i>Pemain menekan tombol level 4</i>
Masukan	Klik tombol <i>level 4</i>
Keluaran yang diharapkan	Pemain tetap berada di antarmuka <i>select level</i>
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain tetap berada di antarmuka <i>select level</i>
<i>Skenario 3</i>	<i>Pemain menekan tombol Reset</i>
Masukan	Klik tombol <i>Reset</i>
Keluaran yang diharapkan	Menampilkan menu konfirmasi <i>reset level</i>
Hasil uji coba	Berhasil
Kondisi Akhir	Menu konfirmasi <i>reset level</i> tampil
<i>Skenario 3.1</i>	<i>Pemain menekan tombol Yes pada konfirmasi reset</i>
Masukan	Klik tombol <i>Yes</i>
Keluaran yang diharapkan	<i>Reset progres level</i> dilakukan. <i>Level 2</i> dan <i>3</i> kembali tertutup
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain mengulang progres permainan dari awal kembali
<i>Skenario 3.2</i>	<i>Pemain menekan tombol No pada konfirmasi reset</i>
Masukan	Klik tombol <i>No</i>
Keluaran yang diharapkan	Menu konfirmasi reset <i>level</i> ditutup
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada di antarmuka <i>select level</i>
<i>Skenario 4</i>	<i>Pemain menekan tombol Back</i>
Masukan	Klik tombol <i>Back</i>

Keluaran yang diharapkan	Pemain berpindah ke antarmuka menu utama
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada di antarmuka menu utama

5.2.5 Uji Coba Pada Antarmuka *Game*

Uji coba fungsionalitas yang akan dilakukan pada antarmuka *game* adalah skenario yang terjadi ketika menekan tombol-tombol *Hero*, tombol *Pause*, dan skenario menang kalah. Uji coba pada antarmuka ini dilakukan dengan skenario antarmuka *level 1* dalam keadaan tombol *hero 1* aktif, tombol *hero 2* *cooldown*, tombol *hero 3* tidak aktif karena *gold* kurang, sedangkan tombol 4 sampai 7 terkunci. Skenario uji coba tersebut dapat dilihat pada Gambar 5.3. Hasil uji coba antarmuka login dapat dilihat pada Tabel 5.5.



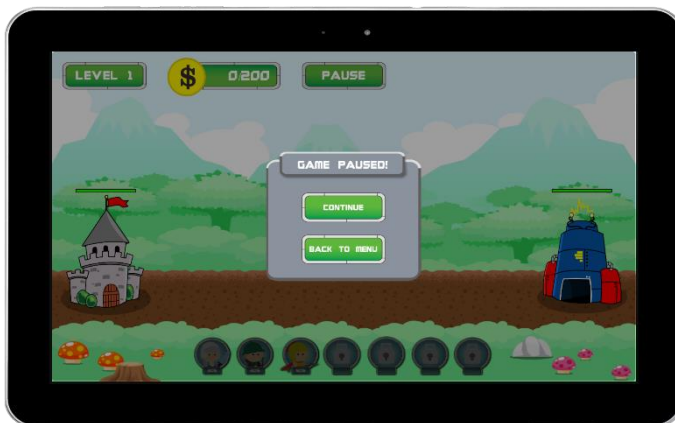
Gambar 5.3 Skenario uji coba antarmuka *game*

Tabel 5.5 Hasil uji coba antarmuka *game*

ID	UF-005
Nama	Uji Coba Pada Antarmuka <i>Game</i>
Tujuan uji coba	Pemain mengetahui fungsionalitas tombol-tombol pada <i>game</i>
Kondisi awal	Pemain berada pada antarmuka <i>level 1</i> , tombol <i>hero 1</i> aktif, tombol <i>hero 2</i> <i>cooldown</i> , tombol <i>hero 3</i> tidak aktif, tombol 4 sampai 7 terkunci.
<i>Skenario 1</i>	<i>Pemain menekan tombol Hero 1</i>
Masukan	Klik tombol <i>Hero 1</i>
Keluaran yang diharapkan	Pemain membeli <i>hero 1</i> dan mengirimnya ke medan pertempuran
Hasil uji coba	Berhasil
Kondisi Akhir	Uang pemain berkurang dan <i>hero 1</i> muncul
<i>Skenario 2</i>	<i>Pemain menekan tombol Hero 2</i>
Masukan	Klik tombol <i>Hero 2</i>
Keluaran yang diharapkan	Pemain tidak dapat membeli <i>hero 2</i> karena sedang dalam keadaan <i>cooldown</i>
Hasil uji coba	Berhasil
Kondisi Akhir	Uang pemain tidak berkurang dan <i>hero 2</i> tidak muncul
<i>Skenario 3</i>	<i>Pemain menekan tombol Hero 3</i>
Masukan	Klik tombol <i>Hero 3</i>
Keluaran yang diharapkan	Pemain tidak dapat membeli <i>hero 3</i> karena uang pemain tidak cukup
Hasil uji coba	Berhasil
Kondisi Akhir	Uang pemain tidak berkurang dan <i>hero 3</i> tidak muncul
<i>Skenario 4</i>	<i>Pemain menekan tombol Hero 4</i>
Masukan	Klik tombol <i>Hero 4</i>

Keluaran yang diharapkan	Pemain tidak dapat membeli <i>hero 4</i> karena terkunci
Hasil uji coba	Berhasil
Kondisi Akhir	Uang pemain tidak berkurang dan <i>hero 4</i> tidak muncul
<i>Skenario 5</i>	<i>Pemain menekan tombol Pause</i>
Masukan	Klik tombol <i>Pause</i>
Keluaran yang diharapkan	Menampilkan menu <i>paused game</i>
Hasil uji coba	Berhasil
Kondisi Akhir	Menu <i>paused game</i> tampil
<i>Skenario 5.1</i>	<i>Pemain menekan tombol Continue</i>
Masukan	Klik tombol <i>Continue</i>
Keluaran yang diharapkan	Menu <i>paused game</i> ditutup
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain kembali ke antarmuka <i>level 1</i> dan melanjutkan permainan
<i>Skenario 5.2</i>	<i>Pemain menekan tombol Back to Menu</i>
Masukan	Klik tombol <i>Back to Menu</i>
Keluaran yang diharapkan	Pemain berpindah ke antarmuka <i>select level</i>
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berpindah ke antarmuka <i>select level</i>

Skenario 5 pada saat pemain menekan tombol *Pause* dapat dilihat pada Gambar 5.4.



Gambar 5.4 Skenario *game paused*

5.3 Pengujian Alur Permainan

Pengujian alur permainan adalah pengujian yang dilakukan untuk mengetahui apakah setiap *level* pada permainan memungkinkan untuk diselesaikan. Pada pengujian ini terdapat dua skenario, yaitu skenario menang dan skenario kalah. Skenario 1 pada saat pemain menghancurkan *tower* musuh atau *win game* dapat dilihat pada Gambar 5.5. Skenario 2 pada saat *tower* pemain hancur atau *lose game* dapat dilihat pada Gambar 5.6. Hasil pengujian alur permainan dapat dilihat pada tabel 5.6.

Tabel 5.6 Hasil pengujian alur permainan

ID	UF-006
Nama	Uji Coba Alur Permainan
Tujuan uji coba	Pemain mengetahui kondisi menang dan kondisi kalah
Kondisi awal	Memulai permainan
<i>Skenario 1</i>	<i>Pemain menghancurkan Tower musuh (kondisi menang)</i>

Masukan	-
Keluaran yang diharapkan	Menampilkan menu <i>win game</i>
Hasil uji coba	Berhasil
Kondisi Akhir	Menu <i>win game</i> tampil
<i>Skenario 1.1</i>	<i>Pemain menekan tombol Next Level</i>
Masukan	Klik tombol <i>Next Level</i>
Keluaran yang diharapkan	Pemain berpindah ke antarmuka <i>level 2</i>
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada di antarmuka <i>level 2</i>
<i>Skenario 1.2</i>	<i>Pemain menekan tombol Back to Menu</i>
Masukan	Klik tombol <i>Back to Menu</i>
Keluaran yang diharapkan	Pemain berpindah ke antarmuka <i>select level</i>
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada di antarmuka <i>select level</i>
<i>Skenario 2</i>	<i>Tower pemain dihancurkan (kondisi kalah)</i>
Masukan	-
Keluaran yang diharapkan	Menampilkan menu <i>lose game</i>
Kondisi Akhir	Menu <i>lose game</i> tampil
<i>Skenario 2.1</i>	<i>Pemain menekan tombol Retry</i>
Masukan	Klik tombol <i>Retry</i>
Keluaran	Pemain kembali ke antarmuka <i>level 1</i>
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada di antarmuka <i>level 1</i> dan mengulangi permainan
<i>Skenario 2.2</i>	<i>Pemain menekan tombol Back to Menu</i>
Masukan	Klik tombol <i>Back to Menu</i>

Keluaran	Pemain berpindah ke antarmuka <i>select level</i>
Hasil uji coba	Berhasil
Kondisi Akhir	Pemain berada di antarmuka <i>select level</i>



Gambar 5.5 Kondisi menang game



Gambar 5.6 Kondisi kalah game

5.4 Pengujian Tingkat Kesulitan

Pengujian tingkat kesulitan adalah pengujian yang dilakukan untuk mengetahui apakah tingkat kesulitan pada setiap *level* semakin meningkat mulai dari *level* rendah hingga *level* tinggi. Akan terdapat beberapa parameter yang menjadi tolak ukur dari tingkat kesulitan setiap *level*. Hasil pengujian tingkat kesulitan dapat dilihat pada Tabel 5.7.

Tabel 5.7 Hasil pengujian tingkat kesulitan

<i>Level</i>	<i>Spawning Type</i>	<i>Jenis Enemy</i>	<i>Max Enemy</i>	<i>Max Time (minute)</i>
1	1	3	5	1
2	1	3	5	1
3	1	4	7	1
4	1	4	7	1
5	1	5	9	1.5
6	1	5	9 + 1 boss	1.5
7	2	6	13	1.5
8	2	6	13	1.5
9	2	6	16	2
10	2	7	16	2
11	2	7	19	2
12	2	7	19 + 1 boss	2
13	3	7	22	2.5
14	3	7	22	2.5
15	3	7	25	2.5
16	3	7	25	2.5
17	3	7	28	3
18	3	7	28 + 1 boss	3

Terdapat 3 macam *spawning type* atau cara munculnya *monster*, yaitu :

1. Dengan waktu ditentukan secara pasti setiap T sekon dan kelipatannya, sedangkan *monster* yang akan muncul juga ditentukan secara pasti.
2. Dengan waktu ditentukan secara pasti setiap T sekon dan kelipatannya, sedangkan *monster* yang akan muncul akan ditentukan secara acak.
3. Dengan waktu dan *monster* yang akan muncul ditentukan secara acak.

5.5 Pengujian Usability

Pengujian *usability* dilakukan dengan cara melakukan survei ke pemain di sekitar lingkungan Informatika ITS. Survei dilakukan untuk mengukur nilai dalam hal antarmuka, performa, dan hiburan/*fun* dari *game* yang dibuat. Survei dilakukan dengan cara mengisi kuesioner yang dibagikan kepada 5 *responden*. Hasil kuesioner dapat dilihat pada lampiran hasil kuesioner. Rekapitulasi dari hasil kuesioner dapat dilihat pada Tabel 5.8.

5.6 Evaluasi

Subbab ini membahas mengenai evaluasi terhadap pengujian-pengujian yang telah dilakukan. Dalam hal ini, evaluasi menunjukkan data rekapitulasi dari hasil pengujian fungsionalitas dan pengujian alur permainan. Rekapitulasi disajikan dalam tabel evaluasi hasil uji coba yang dapat dilihat pada Tabel 5.9. Dari data yang terdapat pada tabel tersebut, dapat diketahui bahwa *game* yang dibuat telah berjalan sesuai skenario yang diharapkan pada awal perancangan.

Berdasarkan pengujian tingkat kesulitan, dapat disimpulkan bahwa tingkat kesulitan pada *game* menjadi semakin sulit seiring bertambahnya *level*. Berdasarkan hasil kuesioner pada pengujian *usability* dapat disimpulkan bahwa antarmuka permainan sudah

cukup baik dan mudah digunakan. Rekapitulasi evaluasi terhadap uji coba yang dilakukan dapat dilihat pada Tabel 5.10.

Tabel 5.8 Rekapitulasi hasil kuesioner

No	Parameter	TS	KS	CS	S	S S	Rata- Rata
Parameter Antarmuka							
1.	Instruksi permainan / <i>tutorial</i> jelas		1	3	1		3
2.	Antarmuka <i>game</i> mudah untuk dioperasikan		1	1	3		3,4
3.	Tampilan <i>game</i> bagus dan rapih				4	1	4,1
4.	Tampilan <i>game</i> sesuai dengan tema			5			3
Parameter Performa							
5.	Animasi berjalan dengan lancar			4	1		3,2
6.	Perpindahan antar menu berjalan dengan lancar			3	2		3,4
Parameter Kesenangan (<i>Fun</i>)							
7.	Saya memiliki ketertarikan untuk mencoba <i>game</i> ini			3	2		3,4
8.	Saya menikmati <i>game</i> ini		1	3	1		3,4
9.	Saya ingin memainkan lagi <i>game</i> ini		1	4			2,8
10.	Saya penasaran untuk menyelesaikan semua <i>level</i> pada <i>game</i> ini		2	3			2,6
Jumlah			6	29	14	1	3,2

Tabel 5.9 Evaluasi hasil uji coba

ID	Deskripsi	Kemungkinan / Skenario	Perilaku Terlaksana
UF – 001	Uji Coba Pada Antarmuka Awal	Skenario 1	ya
		Skenario 2	ya
UF – 002	Uji Coba Pada Antarmuka Menu Utama	Skenario 1	ya
		Skenario 2	ya
		Skenario 3	ya
		Skenario 4	ya
UF – 003	Uji Coba Pada Antarmuka <i>Tutorial</i>	Skenario 1	ya
		Skenario 2	ya
		Skenario 3	ya
UF – 004	Uji Coba Pada Antarmuka <i>Select Level</i>	Skenario 1	ya
		Skenario 2	ya
		Skenario 3	ya
		Skenario 3.1	ya
		Skenario 3.2	ya
		Skenario 4	ya
UF – 005	Uji Coba Pada Antarmuka <i>Game</i>	Skenario 1	ya
		Skenario 2	ya
		Skenario 3	ya
		Skenario 4	ya
		Skenario 5	ya
		Skenario 5.1	ya
		Skenario 5.2	ya
UF – 006	Uji Coba Alur Permainan	Skenario 1	ya
		Skenario 1.1	ya
		Skenario 1.2	ya
		Skenario 2	ya
		Skenario 2.1	ya
		Skenario 2.2	ya

Tabel 5.10 Rekapitulasi evaluasi

No	ID	Nama	Hasil
1	EA – 001	Fungsionalitas	Berhasil
2	EA – 002 - 01	Alur permainan (skenario 1)	Berhasil
3	EA – 002 – 02	Alur permainan (skenario 2)	Berhasil
4	EA – 003	Tingkat kesulitan	Berhasil
5	EA – 004	<i>Usability</i>	Berhasil

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari tujuan pembuatan *game* dan hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, terdapat pula saran yang ditujukan untuk pengembangan *game* lebih lanjut.

6.1. Kesimpulan

Dalam proses pengerjaan tugas akhir mulai dari tahap analisis, desain, implementasi, hingga pengujian didapatkan kesimpulan sebagai berikut:

1. Aturan main, skenario, dan tingkat kesulitan sudah dapat diimplementasikan pada *game The Legend Guardians* berupa *level-level* yang terdapat dalam permainan.
2. *Game The Legend Guardians* sudah dapat dirancang dan diimplementasikan menggunakan *game engine* Unity.

6.2. Saran

Pada subbab ini akan dijelaskan beberapa saran untuk pengembangan *game* di masa yang akan datang. Berdasarkan pada hasil perancangan, implementasi, dan uji coba yang telah dilakukan, *game The Legend Guardians* dapat dikembangkan menjadi sebagai berikut:

1. Tambahkan *mode multiplayer* pada permainan agar batas pemain menjadi lebih dari 2 pemain. *Mode multiplayer* dapat ditambahkan dengan merancang dan mengimplementasikan arsitektur jaringan, komunikasi data, dan sinkronisasi antar pemain. Dengan demikian, permainan akan menjadi lebih berkembang dan strategis.

2. Jumlah karakter *hero* dalam *game* ditambah lagi, sehingga pemain dapat memilih *hero* terlebih dahulu sebelum permainan dimulai.
3. Terdapat sistem *level* dan *upgrade* pada setiap karakter *hero*, sehingga permainan menjadi lebih seru dan bervariasi.

DAFTAR PUSTAKA

- [1] M. DaGraca, "Practical Game AI Programming", Packt Publishing, 2017. [Online] Available: <https://www.safaribooksonline.com/library/view/practical-game-ai/9781787122819> [Accessed 25 Juni 2018].
- [2] L. Kuan Keat, "Rescue Rangers", 5 Juni 2015. [Online]. Available: <http://sea.ign.com/line-rangers/89997/review/line-rangers-review> [Accessed 25 Juni 2018].
- [3] Richard C. Moss, "The Least Worst Idea We Have – The Creation of the Age of Empires", 1 Agustus 2018. [Online]. Available: <https://arstechnica.com/gaming/2018/01/the-age-of-age-of-empires-as-told-by-the-devs-who-built-it/> [Accessed 25 Juni 2018].
- [4] Annisa U. Istighotsah, I. Kuswardayan, and D. Sunaryono, "Rancang Bangun "Chomical": Game Tower Defense dengan Menerapkan Eksperimen Kimia Menggunakan Kerangka Kerja libGDX", Jurnal Teknik POMITS, vol. 2, no. 2, pp. A312-A315, Sep. 2013.
- [5] A. Hernández Sabaté, Meritxell Joanpere, Núria Gorgorió, Lluís Albarracín, "Mathematics learning opportunities when playing a Tower Defense Game", Research Gate, vol. 2, issue 4, pp. 57-71, Des. 2015.
- [6] W. Li, F. Abtahi, C. Tsangouri, and Z. Zhu, "Towards an "In-the-Wild" Emotion Dataset Using a Game-based Framework." pp. 75–83, 2016.
- [7] E. Burnette, "Hello, Android", Texas: Pragmatic Programmers, 2010.
- [8] Tekno Jurnal, "Unity – Game Engine Tangguh untuk Berbagai Platform", 16 Juli 2014. [Online]. Available: <https://teknojurnal.com/unity> [Accessed 25 Juni 2018].
- [9] L. Mariani, M. Pezze, O. Riganelli, and M. Santoto, "AutoBlackTest: Automatic Black-Box Testing of Interactive Applications".

- [10] I. Kuswardayan, Pengembangan Game Multiplatform, Surabaya: PIKTI, 2016.
- [11] P. Co, Level Design for Games: Creating Compelling Game Experiences, Michigan: New Riders Games, 2006.

LAMPIRAN

HASIL KUESIONER



KUESIONER TUGAS AKHIR – 05111240007004 MUHAMMAD ABDURROHMAN

PERANCANGAN DAN IMPLEMENTASI GAME TOWER DEFENSE MULTI PEMAIN: THE LEGEND GUARDIANS BERBASIS ANDROID

A. Identitas Responden

Nama Lengkap : Imam Afandi Usia : 23 Tahun
Pekerjaan : mahasiswa Jenis Kelamin : L P

B. Penilaian Terhadap Game

Setelah memainkan game, silahkan berikan penilaian untuk setiap poin tabel dibawah ini. Skala penilaian adalah sebagai berikut :

SS = Sangat Setuju S = Setuju CS = Cukup Setuju
KS = Kurang Setuju TS = Tidak Setuju

No	Parameter Antarmuka	TS	KS	CS	S	SS
1.	Instruksi permainan / tutorial jelas			✓		
2.	Antarmuka game mudah untuk dioperasikan				✓	
3.	Tampilan game bagus dan rapih				✓	
4.	Tampilan game sesuai dengan tema			✓		
Parameter Performa						
5.	Animasi berjalan dengan lancar				✓	
6.	Perpindahan antar menu berjalan dengan lancar				✓	
Parameter Kesenangan (Fun)						
7.	Saya memiliki ketertarikan untuk mencoba game ini			✓		
8.	Saya menikmati game ini			✓		
9.	Saya ingin memainkan lagi game ini			✓		
10.	Saya penasaran untuk menyelesaikan semua level pada game ini			✓		

C. Kritik dan Saran

Secara keseluruhan udah bagus mulai dari tampilan
sampai game nya sayang sekali multi player nya belum bisa

Surabaya, 25 juni, 2018

Imam

Gambar 8.1 Kuesioner responden 1



KUESIONER TUGAS AKHIR – 05111240007004 MUHAMMAD ABDURROHMAN

**PERANCANGAN DAN IMPLEMENTASI GAME TOWER DEFENSE
MULTI PEMAIN: THE LEGEND GUARDIANS BERBASIS ANDROID**

A. Identitas Responden

Nama Lengkap : Tika Ramesthanah Usia : 23 Tahun
Pekerjaan : mahasiswa Jenis Kelamin : L (P)

B. Penilaian Terhadap Game

Setelah memainkan game, silahkan berikan penilaian untuk setiap poin tabel dibawah ini. Skala penilaian adalah sebagai berikut :

SS = Sangat Setuju S = Setuju CS = Cukup Setuju
KS = Kurang Setuju TS = Tidak Setuju

No	Parameter Antarmuka	TS	KS	CS	S	SS
1.	Instruksi permainan / tutorial jelas				✓	
2.	Antarmuka game mudah untuk dioperasikan			✓		
3.	Tampilan game bagus dan rapih				✓	
4.	Tampilan game sesuai dengan tema			✓		
Parameter Performa						
5.	Animasi berjalan dengan lancar			✓		
6.	Perpindahan antar menu berjalan dengan lancar			✓		
Parameter Kesenangan (Fun)						
7.	Saya memiliki ketertarikan untuk mencoba game ini				✓	
8.	Saya menikmati game ini				✓	
9.	Saya ingin memainkan lagi game ini			✓		
10.	Saya penasaran untuk menyelesaikan semua level pada game ini			✓		

C. Kritik dan Saran

Karakternya lucu-lucu animasinya juga bagus, cuma masih
kurang banyak variasi, jadi kedepannya ditambah lagi.
.....
.....

Surabaya, 25 Juni, 2018

Gambar 8.2 Kuesioner responden 2



KUESIONER TUGAS AKHIR – 05111240007004 MUHAMMAD ABDURROHMAN

**PERANCANGAN DAN IMPLEMENTASI GAME TOWER DEFENSE
MULTI PEMAIN: THE LEGEND GUARDIANS BERBASIS ANDROID**

A. Identitas Responden

Nama Lengkap : Ridwanul Fata Usia : 29 Tahun
Pekerjaan : mahasiswa Jenis Kelamin : L P

B. Penilaian Terhadap Game

Setelah memainkan game, silahkan berikan penilaian untuk setiap poin tabel dibawah ini. Skala penilaian adalah sebagai berikut :

SS = Sangat Setuju S = Setuju CS = Cukup Setuju
KS = Kurang Setuju TS = Tidak Setuju

No	Parameter Antarmuka	TS	KS	CS	S	SS
1.	Instruksi permainan / tutorial jelas		✓			
2.	Antarmuka game mudah untuk dioperasikan		✓			
3.	Tampilan game bagus dan rapih				✓	
4.	Tampilan game sesuai dengan tema			✓		
Parameter Performa						
5.	Animasi berjalan dengan lancar			✓		
6.	Perpindahan antar menu berjalan dengan lancar			✓		
Parameter Kesenangan (Fun)						
7.	Saya memiliki ketertarikan untuk mencoba game ini			✓		
8.	Saya menikmati game ini		✓			
9.	Saya ingin memainkan lagi game ini			✓		
10.	Saya penasaran untuk menyelesaikan semua level pada game ini		✓			

C. Kritik dan Saran

tampilannya ok, tapi saya kurang paham cara
bermainnya

Surabaya, 25 Juni 2018

Ridwanul Fata

Gambar 8.3 Kuesioner responden 3



KUESIONER TUGAS AKHIR – 0511124007004 MUHAMMAD ABDURROHMAN

**PERANCANGAN DAN IMPLEMENTASI GAME TOWER DEFENSE
MULTI PEMAIN: THE LEGEND GUARDIANS BERBASIS ANDROID**

A. Identitas Responden

Nama Lengkap : Fauzan Andi F. Usia : 23 Tahun
Pekerjaan : Mahasiswa. Jenis Kelamin : (L) P

B. Penilaian Terhadap Game

Setelah memainkan game, silahkan berikan penilaian untuk setiap poin tabel dibawah ini. Skala penilaian adalah sebagai berikut :

SS = Sangat Setuju S = Setuju CS = Cukup Setuju
KS = Kurang Setuju TS = Tidak Setuju

No	Parameter Antarmuka	TS	KS	CS	S	SS
1.	Instruksi permainan / tutorial jelas			✓		
2.	Antarmuka game mudah untuk dioperasikan				✓	
3.	Tampilan game bagus dan rapih				✓	
4.	Tampilan game sesuai dengan tema			✓		
Parameter Performa						
5.	Animasi berjalan dengan lancar			✓		
6.	Perpindahan antar menu berjalan dengan lancar			✓		
Parameter Kesenangan (Fun)						
7.	Saya memiliki ketertarikan untuk mencoba game ini			✓		
8.	Saya menikmati game ini			✓		
9.	Saya ingin memainkan lagi game ini			✓		
10.	Saya penasaran untuk menyelesaikan semua level pada game ini			✓		

C. Kritik dan Saran

game nya terlalu simple dibuat lebih menantang

Surabaya, 26 juni 2018

Gambar 8.4 Kuesioner responden 4



KUESIONER TUGAS AKHIR – 05111240007004 MUHAMMAD ABDURROHMAN

**PERANCANGAN DAN IMPLEMENTASI GAME TOWER DEFENSE
MULTI PEMAIN: THE LEGEND GUARDIANS BERBASIS ANDROID**

A. Identitas Responden

Nama Lengkap : Muh. Jaber
Pekerjaan : pengusaha

Usia : 23 Tahun
Jenis Kelamin : L / P

B. Penilaian Terhadap Game

Setelah memainkan game, silahkan berikan penilaian untuk setiap poin tabel dibawah ini. Skala penilaian adalah sebagai berikut :

SS = Sangat Setuju
KS = Kurang Setuju

S = Setuju
TS = Tidak Setuju

CS = Cukup Setuju

No	Parameter Antarmuka	TS	KS	CS	S	SS
1.	Instruksi permainan / tutorial jelas			<input checked="" type="checkbox"/>		
2.	Antarmuka game mudah untuk dioperasikan				<input checked="" type="checkbox"/>	
3.	Tampilan game bagus dan rapih					<input checked="" type="checkbox"/>
4.	Tampilan game sesuai dengan tema			<input checked="" type="checkbox"/>		
Parameter Performa						
5.	Animasi berjalan dengan lancar			<input checked="" type="checkbox"/>		
6.	Perpindahan antar menu berjalan dengan lancar				<input checked="" type="checkbox"/>	
Parameter Kesenangan (Fun)						
7.	Saya memiliki ketertarikan untuk mencoba game ini				<input checked="" type="checkbox"/>	
8.	Saya menikmati game ini			<input checked="" type="checkbox"/>		
9.	Saya ingin memainkan lagi game ini			<input checked="" type="checkbox"/>		
10.	Saya penasaran untuk menyelesaikan semua level pada game ini			<input checked="" type="checkbox"/>		

C. Kritik dan Saran

tutorial lebih baik di buat pada level awal agar
sebelum permainan dimulai user bisa paham
cara bermain nya

Surabaya, 26 Juni 2018

JmJad

Gambar 8.5 Kuesioner responden 5

(Halaman Ini Sengaja dikosongkan)

BIODATA PENULIS



Muhammad Abdurrohman, biasa disapa Deua. lahir di kota Bandung pada tanggal 29 Juni 1994. Merupakan anak ketiga dari lima bersaudara. Penulis menempuh pendidikan formal di SDN Sarijadi Selatan II Bandung (2000 - 2006), MTs Darul Arqam Putera Garut (2006 - 2009), MA Darul Arqam Putera Garut (2009 - 2012). Pada tahun 2012 penulis memulai pendidikan S1 melalui jalur Kementrian Agama RI di Departemen Informatika Fakultas Teknologi Informasi dan Komunikasi Institut

Teknologi Sepuluh Nopember Surabaya, Jawa Timur. Semasa kuliah penulis aktif dalam organisasi yang berada dibawah Kementrian Agama RI yaitu CSSMoRA ITS. Penulis dapat dihubungi melalui alamat *email rahmanlogic29@gmail.com*.